INAUGURAL-DISSERTATION
zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich-Mathematischen Gesamtfakultät
der
Ruprecht-Karls-Universität
Heidelberg

vorgelegt von
Dipl.-Ing. Katja Daniela Mombaur
aus Hilden

Tag der mündlichen Prüfung: 31. August 2001

Stability Optimization of Open-loop Controlled Walking Robots

Gutachter: Prof. Dr. Dr. h.c. Hans Georg Bock
Prof. Dr. Richard W. Longman

To my parents

Manfred & Inge Metzger

Acknowledgments

This thesis has been prepared at the Interdisciplinary Center for Scientific Computing (IWR) at the University of Heidelberg. I especially wish to thank my advisors Prof. Dr. Dr. h.c. Hans Georg Bock and Dr. Johannes Schlöder for their interest in and their support for my Ph.D. project. Through many inspiring discussions they have strongly influenced the course of my work. It has alway been a great pleasure to be a member of their *Simulation & Optimization* team.
I am indebted to Prof. Dr. Richard Longman of Columbia University, New York, for his support as engineering supervisor of this thesis and for many fruitful discussions about open-loop stable robots during his stays in Heidelberg.
In addition, I wish to thank Prof. Dr. Dr. h.c. Willi Jäger, who supervised my work in the *Graduiertenkolleg Modellierung und Wissenschaftliches Rechnen in Mathematik und Naturwissenschaften*. Travel support by the German Research Council (DFG) within this graduate program is gratefully acknowledged.
Successful scientific research can only be done in a highly motivating and inspiring work environment. For having provided such, I owe many thanks to Dr. Michael Winckler who shared the office with me during the years of my thesis. He has also provided valuable help on integrators, switching functions, object-oriented programming and much more. Many thanks go to Christian Kraus for patiently answering all my questions when I was learning C++. I thank Dr. Daniel Leineweber and Andreas Schäfer for providing MUSCOD and Thorsten Stoßmeister and Jan Simon for many useful suggestions. Besides all those explicitely addressed above numerous colleagues at the IWR have contributed to the results of this thesis by valuable discussions.
Special thanks also go to Margret Rotfuß for her help in many areas and to the group's system administrators Thomas Klöpfer and Oliver Bösl.
I thank Dr. Moritz Diehl and Ulrich Brandt-Pollmann as well as Patrick Mombaur, Tom Argyle and Inge Metzger for proofreading parts of this thesis.
I also would like to thank Dr. Mike Coleman and Prof. Dr. Andy Ruina for their hospitality and cooperation during my stay at Cornell University.
Frédéric Moret is gratefully acknowledged for the creation of my personal robot logo that decorates the pages of this thesis.
My family has always been a big help for me. I wish to thank my parents for their love and continuous support for my scientific career. I especially thank my father for stimulating my interest in engineering and mathematics: he could not finish his own Ph.D. thesis before he died in 1999.
Last but not least I wish to thank my husband Patrick for his love and encouragement during our studies of aeronautical engineering in Stuttgart and Toulouse and during the years of my research for this thesis. Our daughter Chiara who was born 11 1/2 weeks ago and kept me awake at night time was another motivating factor for finishing this thesis.

Contents

Introduction

Research on walking robots - 'walking' here being used as a general term for walking, running and crawling - is today one of the most important areas of research in robotics. Two objectives fuel these efforts: The development of new or improved walking robots and the pure insight gained into the locomotion itself that in return can be used to optimize biological gait.

The advantage of walking machines over wheeled robots is that walking is clearly more flexible and compatible with most types of surfaces, including very rough surfaces, stairs etc. Also, legged machines can more freely choose footholds and move over obstacles. There is hence a high number of existing applications, e.g. robots for the exploration of planets or military demining missions in rough terrains and industrial walking robots used for maintenance in nuclear power plants or fire-fighting in skyscrapers. Currently, two main directions of research are pursued: One aims at increasing speed and allowing fewer legs on relatively smooth surfaces, the other aims at creating completely autonomous robots capable to move on very rough terrain. Our research follows the first path concentrating on fast dynamical walking robots with one or two legs.

The pure insight derived from research into walking robots helps us understand the human gait, its mechanisms, its control and its stability. Optimality studies of human motion can result in improved performances in various sports, and parameter studies will lead to conclusions for pathological gait which has mechanical and not neurological causes.

Dynamical walking robots can either be closed-loop or open-loop controlled. While closed-loop control clearly is the most flexible solution allowing the highest number of applications some significant drawbacks exist: It requires sophisticated and expensive sensory systems and feedback-controllers. The computation of appropriate reactions is time critical and often a limitation for making motion faster, hence requiring high computation capacities on-board. This all translates into the necessity of high budgets and deep technical knowledge.

An open-loop control strategy does not use active reaction to respond to perturbations but entirely relies on the mechanical system's natural kinematics and dynamics to stabilize the trajectory. Actuator histories are a priori determined, prescribed and not changed by any feedback interference. Its outstanding advantages are low cost and speed of control. And even for motions on rough terrain where closed-loop control is a necessity, robust open-loop stable trajectories can provide a basis on top of which closed-loop control is

applied.

Not much research has been done in the field of open-loop controlled robots so far. Typically an intuitive approach has been taken: only simple models have been studied, and the stabilization task has been broken down into a number of basic operations.

In this thesis we take a fundamentally different approach. For the first time the problem of open-loop stabilization is addressed by means of optimization methods. By leaving the intuitive path and focusing on mathematical analysis we are able to treat robot models of increased complexity. The goal of our work is

- to provide efficient optimization procedures for the determination of open-loop stable robot parameters and periodic trajectories and
- by finding previously unknown open-loop stable robots to demonstrate the flexibility of the concept of open-loop control.

The research of this interdisciplinary thesis is thus clearly motivated by the applicational aspect.

Besides open-loop controlled actuated walking robots which are our main interest, we also treat the special case of purely mechanical passive-dynamic walkers that, in addition to lacking feedback control, have no active sources of energy. Describing the motion of both types of robots leads to complex periodic multi-phase problems with discontinuities caused by ground and joint contacts. Actuated and passive-dynamic systems lead to non-autonomous and autonomous differential equations, respectively.

A number of significant contributions have been made during our research and will be described in this thesis.

New open-loop stable robot configurations have been discovered all of which exhibit remarkable features. Among others we present simulations of

- the first human-like actuated open-loop stable robot
- the first open-loop stable actuated one-legged hopping robot with point foot
- the first 3D passive-dynamic walker.

The last robot stems from a cooperation with Coleman [18] from Cornell university who assembled a similar real robot and did the modeling, but was not able to find stable solutions. The other two robots have no real counterparts. The entity of these robots not only serves to illustrate the range of possible open-loop stable mechanisms but it also uncovers previously unknown facts about multibody systems and reveals open-loop features of human gait. A unified approach to gait modeling is introduced. Various animation sequences have been produced for all robots and different types of motion based on the visualization tool JAFV (Winckler [102]). We will show a few sequences in this thesis.

A numerical method for the optimization of open-loop stability of periodic systems will be presented. This is the first time stability optimization is combined with the simultaneous solution of a periodic optimal control problem. Stability is defined in terms of the spectral radius of the monodromy matrix which is non-differentiable and may be non-Lipschitz at points of multiple maximum eigenvalue and involves the computation of sensitivities hence representing a difficult non-standard optimization criterion. We introduce a two-level optimization approach splitting the problems of periodic gait generation and stabilization of the system. For the development of these stability optimization methods we could build upon the extensive knowledge and methods for the solution of optimal control problems available in the research group of Bock & Schlöder at IWR, University of Heidelberg (compare Bock & Plitt [11], Leineweber [48], [47]). We have chosen a modular object-oriented approach for implementation since it allowed us to evaluate different possible methods for the non-standard task of stability optimization. A direct search method, which is a modification of the Nelder-Mead polytope algorithm, has shown to be a very good choice. Apparently new equations for the derivatives of the monodromy matrix in the presence of discontinuities (representing second order derivatives of the dynamics) and for the derivatives of singular values are given in this thesis. A numerical criterion for the characterization of nonlinear stability properties will be introduced.

We give general recommendations on the use of alternative objective functions for stability optimization based on extensive theoretical and numerical studies. We have evaluated the use of matrix norms instead of the spectral radius since they represent its upper bounds. Instead of the monodromy matrix a power thereof can be used. These alternative optimization criteria have the advantage of leading to problems easier to solve than eigenvalue optimization.

This thesis is organized in nine chapters and one appendix. Due to the interdisciplinary setting of this work, chapters have been written with different focus on robotics, numerical mathematics, and software engineering.

Chapter 1 serves to motivate the idea of open-loop control as the central topic of this thesis. Its advantages are illustrated against the background of conventional control concepts. Passive-dynamic walking machines are introduced as a special form of open-loop stable robots. Stability properties of different classes of mechanical systems are recalled. We outline our two-level optimization approach to the question of finding open-loop stable robot configurations.

Chapter 2 is dedicated to modeling periodic gaits in robotics and biology. In the first part of the chapter we introduce the general form of periodic multiphase gait models. We give reasons for a prescription of the order of phases. In the second part we describe the modeling process starting from a physical robot model, choosing an appropriate set of coordinates and setting up the equations of motion.

Chapters 3 - 6 describe mathematical background and numerical methods required for the stability optimization of periodic gaits.

In chapter 3 we address the problem of periodic gait generation which is the task to be solved in the inner loop. We recall terminology from the field of dynamical systems and basic theory about existence and uniqueness of solutions of periodic boundary value problems. We give the full formulation of a standard multi-phase optimal control problem with discontinuities. The numerical solution of periodic optimal control problems by a structured direct multiple shooting approach is described.

In chapter 4 we present the mathematical stability criterion to be used in the outer loop to define the stability of the inner loop solution. We recall Lyapunov's first method for differential equations with periodic right hand side being an extension of Floquet theory to nonlinear systems. We show that Lyapunov's first method may be generalized to periodic multi-phase problems with discontinuities.

In chapter 5 we discuss the various difficulties of using stability in terms of the spectral radius of the monodromy matrix as optimization criterion. Possible alternatives are discussed replacing the spectral radius or the monodromy matrix (or both). We list the different optimization criteria to be evaluated and compared in this thesis.

Chapter 6 is dedicated to numerical methods for stability optimization. We start with a brief review of literature in the field of non-smooth optimization and eigenvalue optimization. We describe the direct search method, a variant of the Nelder-Mead polytope algorithm, that we have used for stability optimization of all our robot examples. New formulas for the computations of derivatives of singular values are given. We describe the computation of the monodromy matrix in the presence of discontinuities and the necessary projections for monodromy matrices of autonomous systems. Previously unpublished formulas for second-order derivatives of discontinuous differential equations with respect to initial values and parameters are derived. Finally we present a numerical procedure for the determination of nonlinear stability margins.

Chapters 7 - 9 are dedicated to three specific open-loop stable walking robots and are probably the most interesting for readers who are especially interested in the mechanical and robotics aspect of this thesis. In chapter 7, we present the one-legged actuated hopping robot. The human-like actuated biped walker is presented in chapter 8. The passive-dynamic Tinkertoy robot is subject of chapter 9. We give the full description of all robot models and extensive results of stability optimization using different optimization criteria.

The final chapter contains a summary of the key results and methods produced in this thesis and a discussion of possible extensions and further research.

Software engineering aspects of our work are presented in the appendix.

Chapter 1

Open-loop Stable Walking Robots

In the scientific community the term 'walking robots' denotes any machine moving on legs. Precisely though, 'walking' is a dynamic form of locomotion where at any instant at least one leg is in contact with the ground. It must therefore be distinguished from 'crawling' which describes a quasi-static motion without the need to balance, and from 'running' which in contrast to walking also involves flight phases. In this thesis we will analyze both walking and running robots.

The first ancestors of today's walking robots were designed – and some of them also manufactured – in the 19th century. They had the form of modified wheels with feet attached or of legged vehicles (see Thring [89]) or were inspired by human or animal-like forms (e.g. horses) and powered either by steam or mechanically by an operator. The recent developments in walking machines as we know them today started in the 1960s in Japan where much of the progress since then has been made. Other important steps of robotics development took place in the United States and in Russia. Europe joined the efforts only quite recently 15 years ago. Today there is a variety of walking robots throughout the world. A very extensive and up-to-date survey of state-of the-art walking robots as well as of their history can be found in the Walking Machine Catalogue of Berns [7].

Our main interest in this chapter and throughout this thesis is the stability of walking robots. We focus on the stability properties of different robot types and the efforts necessary to control their motion. The primary purpose of this chapter is to motivate the idea of open-loop controlled walking.

Section 1.1 serves to illustrate the two major stability concepts used in contemporary robotics, static stability on one hand, and dynamic stabilization relying on active feedback interference or closed-loop control on the other hand. In section 1.2 the fundamentally different approach of open-loop control is introduced, being the central topic in this thesis. A special class of open-loop controlled mechanisms without actuation, the passive-dynamic walking machines, is presented in section 1.3. Section 1.4 gives an overview of stability implications of general properties of mechanical systems. In section 1.5 we show

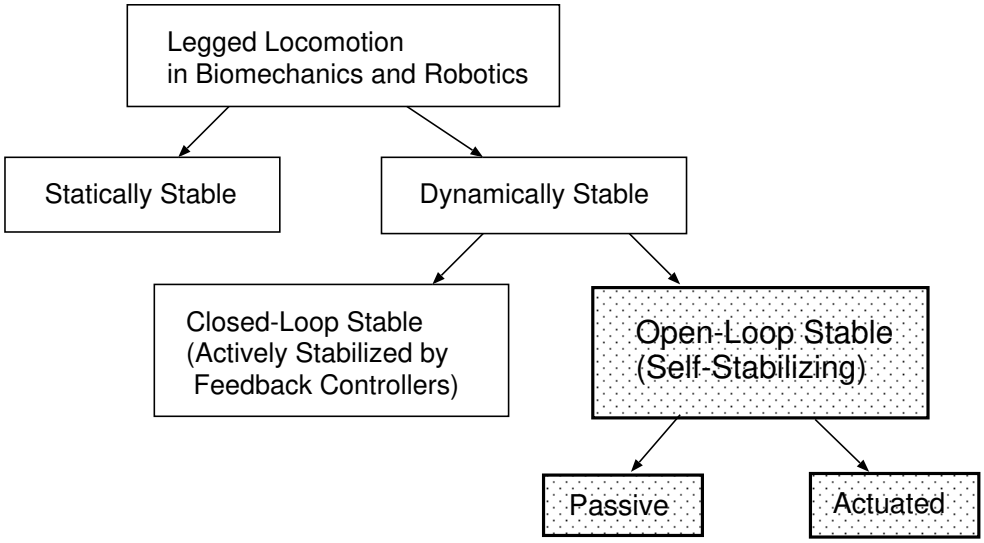


Figure 1.1: Different stability concepts for locomotion

how we approach the problem of finding open-loop stable robot models and configurations by means of optimization.

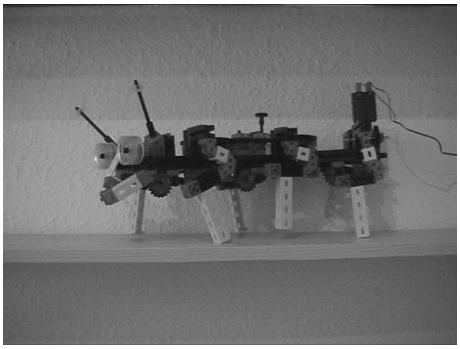
1.1 Common Stability Concepts for Walking Robots

Figure 1.1 illustrates different concepts of stability and stabilization for locomotion. In this section we review the two standard approaches most common among existing robots: statically stable walking, and dynamically stable closed-loop controlled walking. The treatment of the two highlighted concepts in figure 1.1, open-loop controlled actuated and passive walking, which are the focus of this thesis, is deferred to the next two sections.

1.1.1 Statically Stable Walking

Statically stable walking is also referred to as crawling according to the above definition. Natural locomotion of many insects falls into this category (when not flying).

Animals and robots moving in a statically stable fashion do not need to actively balance. Their center of mass (c.o.m.) which is also referred to as zero moment point in robotics always lies within the polygon of support of their stance legs. This necessitates a 3-point ground contact at any instant, generally realized by three feet on the ground. Therefore statically stable walking robots theoretically need at least four legs, but as four legs with only one lifted at a time leads to an awkward gait, in practice they typically have six legs or more. Figure 1.2 shows the typical statically stable tripod gait of a six-legged robot: fore and hind leg of one side together with the middle leg of the other side are lifted,



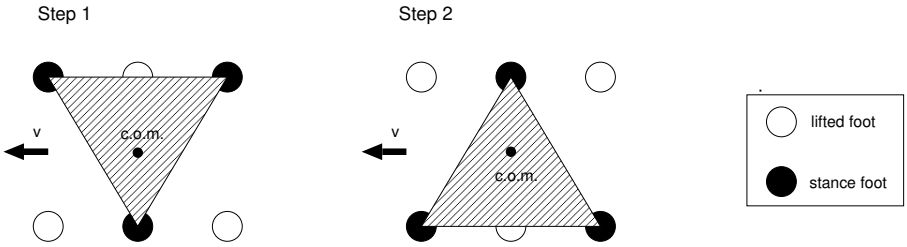


Figure 1.2: Typical tripod gait of statically stable six-legged robots

alternating sides from step to step.

A tricky way to overcome the minimum-number-of-legs requirement sometimes applied is to use less, e.g. two, but very large feet such that a single foot spans an area of support large enough to provide static stability (figure 1.3).

Another requirement for a gait being statically stable is low speed. If the momentum was too high, the c.o.m. might be driven out of the polygon of support and the robot would risk to tip over.

Most walking robots built in the nineties fall into the category of statically stable robots.

Figure 1.3: Statically stable two-legged robots with large feet

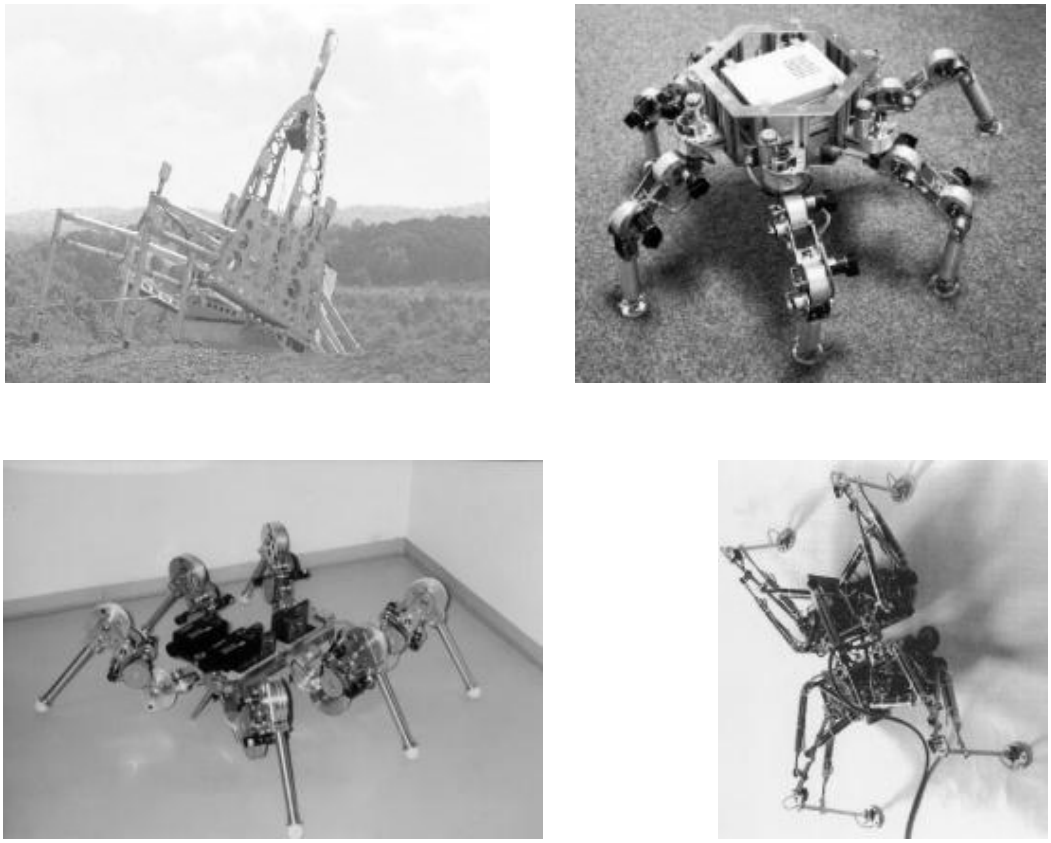


Figure 1.4: Statically stable robots with eight legs (a) Dante – Carnegie Mellon University) and six legs (b) Katharina – Universität Magdeburg, c) Lauron – FZI Karlsruhe, d) RobugIIs – University of Portsmouth). All Pictures taken from the Walking Machine Catalogue of Berns [7]

They have been built for a number of special applications: planetary exploration, maintenance jobs in hazardous areas, demining, forestry etc. (compare e.g. the proceedings [104] and [105]). They have become increasingly sophisticated, use complex sensors and control systems like neural networks to navigate and state-of the art motors, but from a stability point of view they basically rely on the simple concept of static stability. Much of the recent progress has been inspired by the observation of legged locomotion in nature, e.g. of insects. The goal is to develop completely autonomous walking machines.

Related to those statically stable walking machines are climbing robots which are additionally equipped with some sort of suction cups on their feet that enable them to climb up vertical walls. They are used e.g. in skyscrapers for fire-fighting and window cleaning.

Some examples of statically stable walking and climbing robots in use today are given in figure 1.4.

1.1.2 Dynamically Stable Walking with Closed-loop Control

Systems that do not have enough legs or move too fast to satisfy the above static stability requirements can only walk in a dynamically stable or dynamically stabilized fashion. They are facing the problem of balance in every step. A dynamically moving stable system must not fall down, but it must be allowed to tip for short intervals if adequate support is provided afterwards. Locomotion of humans and most mammals belongs to this type of walking.

The definition of dynamic stability is sometimes blurred in the walking machines community, and one has to be careful not to be mislead by the different definitions of dynamic stability. E.g. Karčnik et al. [43] define dynamic stability as the ability of the system to stop within one step. Vukobratovic et al. [97] introduce the definition of practical stability of dynamic systems that is based on three sets of allowed states for initial, end and intermediate configurations.

We define dynamic stability according to the small-perturbations-definition that is most common: a motion of a walking robot is dynamically stable if it persists even in the presence of small perturbations. We will discuss later what small means in this context.. For a robot with closed-loop control this is possible if the controllers take appropriate measures to eliminate the effect of external perturbations and bring the robot back to its original trajectory. The formal mathematical definitions expressing this physical property will be revised in chapter 4. Sometimes not all variables are relevant for the stability of a gait in the sense that perturbations in some directions do not have to be eliminated (e.g. perturbations in the direction of travel for walking on level ground).

One of the first to study dynamically stable walking was Raibert from MIT. The MIT Leg Lab has produced an amazing collection of dynamical walking and running robots over the years (see its homepage [45]). They move at different speeds and in different gaits, some of them are able to get over obstacles or to climb stairways, and they all rely on a common set of balance and control principles.

Of course there are dynamically walking robots and animals with more than two legs. Every possible form of gait of horses, cats or dogs has phases with less than three feet on the ground, even the slow pace of a horse. But in this section we will concentrate on some examples of dynamically stable one- and two-legged machines for comparison with robots treated later in this thesis. Please note that the overview is by no means complete.

Hopping is the only possible form of motion for one-legged robots, and it is equivalent to running according to the definitions at the beginning of this chapter.

One of the first robots built to explore the problem of balance was the hopping monopod of Raibert & Sutherland [75] which moves like a kangaroo or pogo stick. It consists of a toroidal body and a leg which are connected by a hinge powered by a torque. The leg bounces on an adjustable spring. Following the basic idea of breaking the control task down into three independent parts of height, balance and attitude control the motion is controlled by three independent servo-control loops. This robot is related to the open-

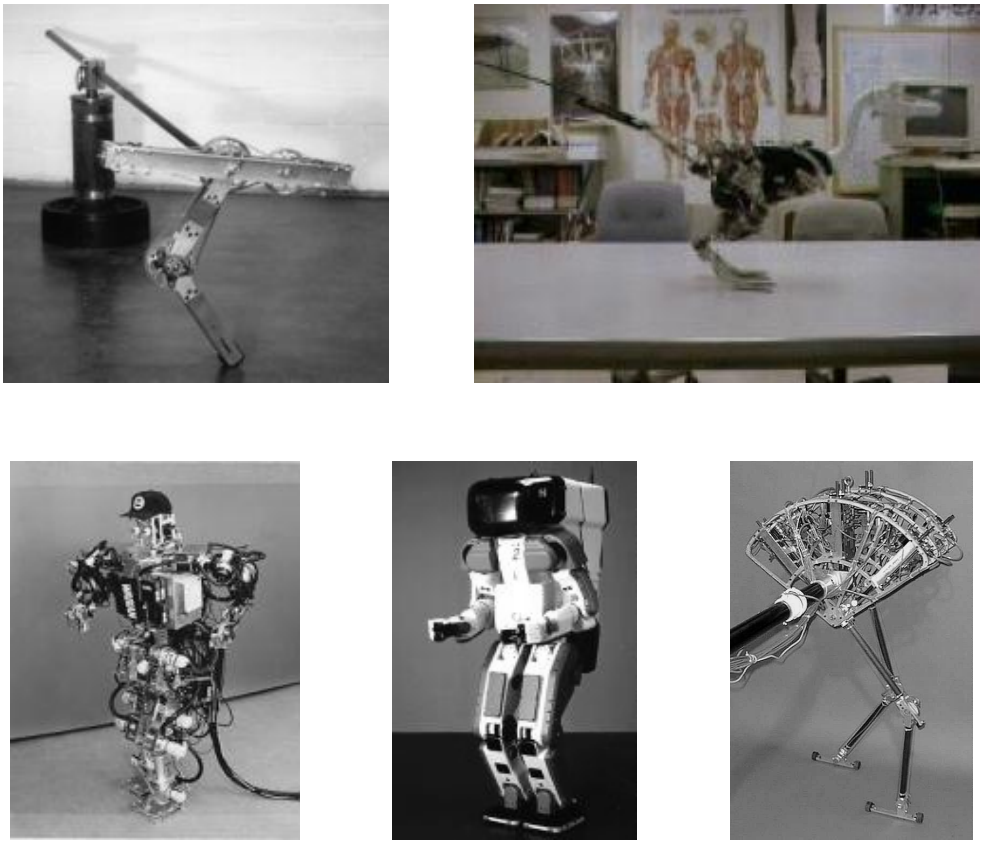


Figure 1.5: Dynamically stable robots with one and two legs (a) OLIE – Vrije Universiteit Brussel, b) Troody – MIT, c) Wabian – Waseda University, d) P2 – Honda Motor Co., e) Spring Flamingo – MIT, all pictures taken from the Walking Machine Catalogue of Berns [7])

loop hopping robot studied in this thesis. Raibert also built a 3D extension of this 2D machine and proposed related multi-legged versions. Another one-legged hopping robot is OLIE of de Man et al. [23] (figure1.5a) that has an articulated instead of a springy leg. Its motion is restrained to a 2D cylindrical surface by a boom and relies on sensors measuring hip position and orientation of all limbs and detecting ground contact.

The first bipedal robots have been built in Japan in the sixties, where researchers at Waseda university built a series of robots. The first of those were static walkers performing very slow gaits (2 min./step!). Their later models were significantly improved as far as walking speed and dynamical balancing abilities are concerned. The number of degrees of freedom and the complexity of the control system have been increased over the years. The most recent model is WABIAN (see figure 1.5c). All Waseda robots rely on playing back pre-recorded trajectories for trunk, legs and arm motions.

Probably the most ambitious and costly bipedal robot project in the world is the Honda Humanoid Robot Project (see their internet page [17]). The goal is to develop an intelligent mobile service robot for general purpose home use. The first prototype P1 was ready after 10 years of development and is said to have consumed 300 person years and 1 million Dollar worth of parts. The current models are P2 (see figure 1.5d) and its lighter successor P3.

A number of bipedal walking machines were built at the MIT Leg Lab [45]. The goal was to develop devices that walk fast and efficiently, are reliable and have large margins of stability. Recent examples are Spring Flamingo of Pratt [74] (see figure 1.5e), walking in the sagittal plane, Troody, a three-dimensional bipedal dinosaur robot with a tail to provide balancing support (figure 1.5b), and the more human-like 3D biped M2. Building and controlling these robots was only possible through enormous technical experience collected during the work on their predecessors.

Although closed-loop control clearly is the most flexible solution we want to point out some important draw-backs that should not be forgotten:

- closed-loop control typically requires sophisticated and expensive sensory systems and feedback-controllers, necessitates a high budget and appropriate technical knowledge,
- computation of appropriate reactions is time-critical and is often a limitation for making some motion faster,
- enough computational power has to be provided on-board or the robot has to be restrained to walking on a wire-lace.

In the next two sections we will explain control principles that may be helpful to overcome some of the difficulties associated with closed-loop control.

1.2 The Idea of Open-loop Controlled Walking

Fundamentally different to the discussed concept of active stabilization using feedback-control we now focus on the idea of open-loop control or self-stabilization.

Open-loop stable control strategies do not use active reaction to respond to perturbations but rely instead on the system's geometry and the kinematics and dynamics of motion to stabilize the trajectory. In contrast to closed-loop control there is no need for sensors nor for any on-line computations.

The actuator histories are a priori determined, prescribed for a motion and not changed by any feedback interference. If an open-loop stable system is slightly perturbed, it will recover without any modification of the input. It always has to stay synchronized with the exciting frequency that rigorously dictates the phase.

For example open-loop controlled human walking is characterized by the use of just the skeleton and the muscles but neither brain nor senses.

Open-loop stable walking is only possible for adequately selected robot configurations and trajectories. It is the goal of this thesis to determine robot models, parameters and actuations that lead to self-stabilizing motions. As open-loop control requires prescribing a motion it only makes sense to look at regular, i.e. strictly periodic gaits. Handling of unforeseen events, like the necessity to climb over large obstacles or to choose irregular footholds, is of course not possible by pure open-loop control.

Open-loop control has the following advantages:

- For selected systems and operations open-loop control is a cheap and fast control possibility.
- It can be used as a basis on top of which closed-loop control is applied. Systems with improved open-loop performance are more robust, less sensitive to sensor readings and require less feedback effort.
- Understanding open-loop control may help understanding learning control.

Pratt has shown in his thesis [74] about the aforementioned Spring Flamingo that exploiting natural dynamics or self-stabilizing properties does reduce the closed-loop control effort.

A number of self-stabilizing effects for mechanical systems are known:

- A static tinkertoy is rocking about and converging towards its stable upright position because of its low center of mass (lower than center of foot curvature arc). The same simple trick also affects the dynamic stability of systems with curved feet, although the dependence is not that straightforward.
- Withdrawing energy from the system can also serve to damp out perturbations.

- A rolling disk or coin is stable due to precision (Greenwood [35]). The same effect stabilizes the motion of bicycles. Spin plays an important role for stabilization. It is, however, no possible solution for robots walking in the conventional sense, i.e. swinging legs back and forth.
- One famous basic mechanical example is the upright inverted pendulum (a simple pendulum as well as an n-pendulum) that can be open-loop stable if excited by a harmonic oscillation in longitudinal direction (see e.g. Otterbein [67]).
- Hubbard [41] shows that a skate-board is stabilized due to coupling between the rider's rolling angle and the skate-boards steering angle.
- If the neutral point of an airplane lies behind its center of mass, aerodynamic flutter oscillations will naturally be damped out (e.g. Dinkler [26]).

Schaal & Atkeson [81] have studied open-loop controlled robot juggling. They investigated different juggling tasks and found stable solutions for some and improved unstable solutions (that were better starting points for closed-loop controllers) for others. Some of those juggling systems allowed intuitive solutions, such as complete absorption of energy after each cycle or simple geometry variations.

Open-loop control of walking robots is still an open field of research. With this thesis a number of break-throughs were achieved in the following topics:

1. Automation of search for open-loop stable robot configurations and solutions:
   In our work we leave the intuitive approach favored by the authors cited below and focus on mathematical analysis. While some numerical recipes for analyzing the stability of given walking motions were known in the walking robot community, we felt that there clearly was a lack of fast and reliable numerical methods for the generation of new open-loop stable gaits for models of increased complexity. The goal of our research was therefore to develop such methods to be applied to very general robot models.
2. Determination of fundamentally new open-loop controlled robot models:
   Questions of general interest are if open-loop controlled walking is possible in 3D, or if humanlike walking can be self-stabilizing. Typically these models are too complex to allow an intuitive approach and thus clearly require a numerical approach.
3. Improve stability for already existing open-loop stable robots:
   The methods developed in 1. can be used to find sets of parameters or motions very different from those already known that lead to significantly improved stability.

There are only very few open-loop controlled physical robots today. Important for our work was the 2D one-legged self-stabilizing hopping robot of Ringrose [76], [77], as it initiated our interest in open-loop controlled walking and served as a good starting example for the development of our numerical methods (see Mombaur et al. [65]). It consists of

a single springy leg with a curved foot and no upper body and is capable of open-loop stable in-place hopping motions. Ringrose also has studied two- and four-legged robots which are just assemblies of several of the above legs. The stability of those robots is to some extent due to a quite large foot radius allowing a trivial stabilization of the system. The investigations are intuitively verified by a separation of the stabilization task into height, pitch and phase stabilization.

An extension of this robot appeared very recently: Wei et al. [99] built a 3D miniature hopping robot, using the same type of leg as the robots of Ringrose but having an additional balance mass with controllable offset. This robot also uses a large foot radius for stabilization.

Other open-loop controlled robots have been developed as extensions of some passive-dynamic walkers that will be presented in the following section. McGeer applied with different success external and internal torques and toe-off impulses to his 2D bipedal straight-legged passive walker [50], [54].

1.3 Passive-dynamic Walking – The Purely Mechanical Approach

Although passive walking robots historically preceded open-loop controlled actuated machines we have chosen the inverse order for presentation as from our point of view passive-dynamic walking is a special case of open-loop controlled walking.

In addition to lacking feedback control that all open-loop controlled systems are characterized by, passive-dynamic walkers also lack all active sources of energy. They are purely mechanical devices walking down slightly inclined slopes that have no actuators but are accelerated by gravity alone. To resume the analogy to human walking: passive-dynamic walking is like a human being only using his skeleton - and neither brain and senses nor muscles. As indicated by the word 'dynamic', only systems that are not moving in a statically stable fashion belong to this category.

Finding passive-dynamic walkers is a considerably simpler task than finding open-loop stable actuated robots: if subject to a perturbation, passive systems have the possibility to take a different amount of time for some operation, and there is no external exciting frequency to which the systems have to synchronize. This possibility of time-shifts can not be underestimated in its positive influence on the existence of stable periodic motion.

Passive-dynamic walking robots belong to the oldest walking machines. The Ruina Lab of Cornell University lists on its internet page [79] a number of old patents for passive walkers dating back as far as 1888. There are also quite a few older and more recent passive-dynamic walking toys (see figure 1.6).

Research about passive-dynamic walking has mainly been motivated by the fact that certain phases of human gait exhibit a very low muscular activity and are therefore nearly

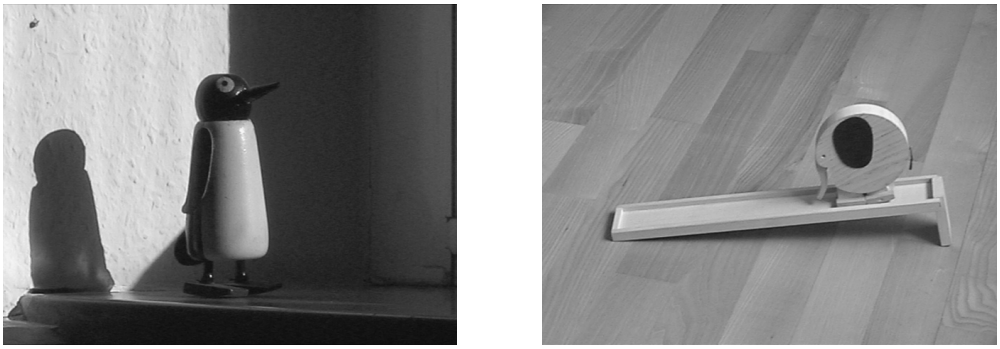


Figure 1.6: Passive-dynamic toy walkers: penguin (1938) and elephant (1998)

passive. It was assumed that mechanical parameters of the human body have a greater influence on human gait than recognized before.

Mochon and McMahon [60], [61] were the first to analyze a passive-dynamic walking model. They showed by comparison with experimental measurements that a kneed walker confined to motions in the sagittal plane with properly chosen parameters can mimic human gait.

The pioneering work in the field of passive-dynamic walking machines was done by McGeer. He studied a series of two-dimensional gravity-powered models of increasing complexity, starting with a rimless wheel, bridging the gap to walking with the synthetic wheel, and finally focusing on stiff-legged and kneed walking devices (McGeer [52], [53], [54]). He recognized the stance leg that is rotating like an inverted pendulum as the key element in passive walking. He also investigated passive-dynamic running of a 2D stiff-legged walker the legs of which were equipped with torsional and longitudinal springs (McGeer [51]). McGeer determined periodic gaits for his sometimes simplified and linearized models using numerical methods and performed linear stability analysis. As mentioned above he also investigated extensions of his passive walkers to actuated open-loop controlled walkers. He applied simple feedback controllers in the cases where no stable solutions could be found. He also built physical models of the bipedal walking robots which he confined to planar motions by adding a third leg next to the others and letting the two outer legs act like a pair of simultaneous crutches. McGeer started to work on a three-dimensional stiff-legged walker but was not able to find stable gaits (McGeer [53]).

The article of Thomson & Raibert [88] is also sometimes cited in the context of passive-dynamic walking machines. They studied a one-legged hopping robot which is passive in the sense that it has no actuators but, as the authors point out themselves, it is not stable, so it does not belong to the class of robots discussed here.

Ruina and his co-workers extended the work of McGeer in various directions. They studied extensively stability as well as chaos of 2D straight-legged walking with point feet. Besides they imitated McGeer's two-dimensional kneed walker in theory and practice

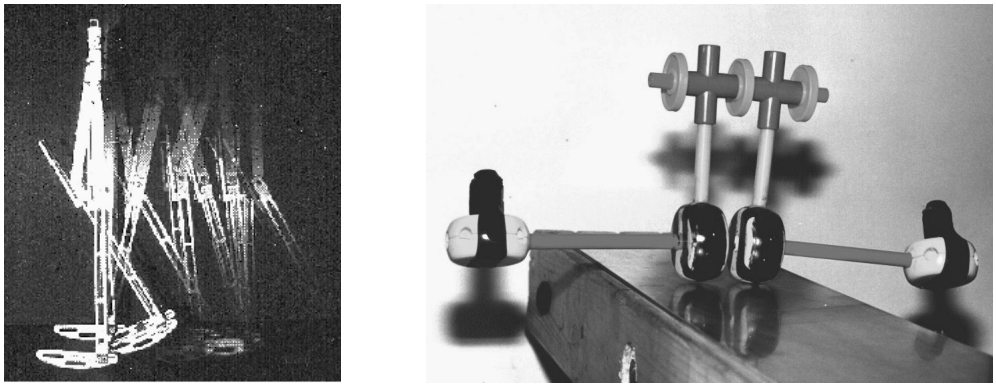


Figure 1.7: Passive-dynamic walking robots of the Ruina Lab [79]

(figure 1.7a) and extended his analysis with special focus on efficency, speed and stability and considering the limit case of near zero slopes (Garcia et al. [31]). Coleman has investigated a series of wheels and walking models following the example of McGeer but with a clear focus on three-dimensional motion and the ultimate goal to find stable walking of a 3D model (Coleman [18]). His search was crowned with the discovery of a simple and fascinating three-dimensional physical toy robot that was obviously capable of stable tinkering gait (figure 1.7b, Coleman & Ruina [20], Coleman [18]). However, all theoretical computations for models related to the physical toy resulted in unstable solutions (see the two previous references and Garcia [30]). Only recently and for the first time, in a collaboration of the author of this thesis with Coleman, Ruina and Garcia it was possible to demonstrate stable 3D walking of a theoretical rigid body model (see Coleman et al. [19]). Further results are given in Mombaur et al. [63] and in chapter 9 of this thesis.

Adolfsson et al. [1] also claim to have found three-dimensional passive-dynamic walking with knees in simulation, but the feet of their model have line contact in lateral direction which prevents it from falling sideways.

The contribution of this thesis in the area of passive-dynamic walking is that our numerical methods developed for general actuated walking devices can equally well be applied to the non-actuated type as special case with all actuations equal to zero. This will be demonstrated on the example of the 3D tinkertoy.

1.4 Stability Properties of Different Types of Mechanical Systems

There are some general qualities of mechanical systems that influence the stability of motion. In this section we will describe the most important ones. In later chapters, when treating specific robot models, we will indicate how they are classified according to this scheme.

We must first introduce some definitions (see e.g. Greenwood [35]) necessary for the characterization of systems.

A system is called *conservative* if its total energy, i.e. the sum of potential and kinetic energy is constant. Conservative systems are characterized by work-less constraints and conservative or non-dissipative forces (like e.g. spring forces). In principle, all mechanical processes in a conservative system can be reversed. In analogy, a system is called *non-conservative*, if its total energy decreases in time. Energy is lost due to non-conservative or dissipative forces or effects like damping, friction or inelastic collisions.

A system is called *holonomic* if there are no constraints at all or if all its constraints are holonomic, i.e. depend only on position variables or generalized coordinates. Holonomic constraints could in theory be used to solve for the independent generalized coordinates.

Non-holonomic systems also have non-holonomic constraints which need to be expressed in terms of differentials of the generalized coordinates and possibly time. These equations are not integrable such that they cannot be used to eliminate coordinates. Non-holonomic systems always require more coordinates for their description than there are degrees of freedom. Probably the most famous example of a non-holonomic system is a disk rolling on a plane.

Walking robots often are characterized by piecewise holonomic, but overall non-holonomic motions. E.g. a biped walker with point feet is holonomic if just the period of one step is considered, and the number of equations necessary to describe the motion during the step is equal to the number of degrees of freedom. After heelstrike and switching of stance and swing leg, the robot is transferred to a region which is not any more accessible by this set of coordinates. So the intermittent contact has introduced a discrete non-holonomy into the system.

Many simple mechanical systems, e.g. the undamped pendulum, are Hamiltonian, i.e. conservative and holonomic. It follows from Liouville's theorem (see e.g. Bronstein et al. [13]) that Hamiltonian systems cannot be asymptotically stable. Any change of energy introduced by a small perturbation persists, it is neither amplified nor damped. If the system is conservative but non-holonomic it can very well be asymptotically stable. The same is true for conservative, piecewise holonomic but overall non-holonomic systems as Ruina [78] has recently demonstrated by an example. It is well known that dissipation can help to promote stability. However, only introducing damping is of course not sufficient to make an arbitrary system stable. But we can summarize that all non-conservative systems - may they be holonomic, piecewise holonomic or non-holonomic - can be asymptotically stable. Coleman [18] gives a more detailed overview on this subject.

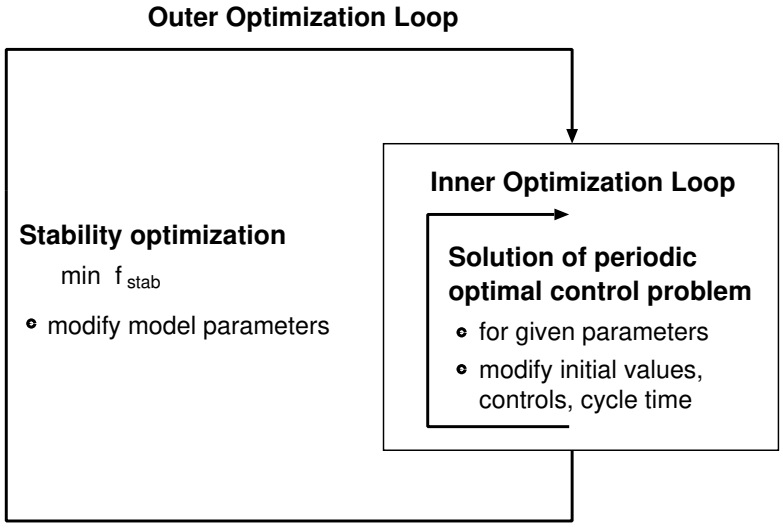


Figure 1.8: Finding open-loop stable robots by means of a two-level optimization procedure

1.5 Finding Open-loop Stable Robots by Means of Optimization

The goal of this thesis is to determine robot models, parameters and, for the active case, actuations that lead to open-loop stable motions. We approach this problem by means of optimization. The development of efficient numerical optimization methods to be applied to arbitrary models is a core component of this thesis.

We have chosen to use a two-level optimization approach that splits the two problems of improving open-loop stability and generating periodic gaits. Figure 1.8 shows a sketch of this two-level optimization procedure. In the outer loop a stability optimization is performed with the model parameters left free for variation. In the inner loop the model parameters are fixed to the values given by the outer loop. A periodic optimal control problem is solved for which controls, initial values of trajectory and periodic cycle time are free variables.

Please note that it is crucial to solve in the inner loop a periodic optimal control problem and not just some periodic boundary value problem. As we will see in chapter 3 the solution of nonlinear periodic boundary value problems is not unique - neither for passive nor for actuated systems. Ignoring this fact might lead to misinformation of the outer loop by the inner loop and thus to failure of the outer loop optimization algorithm. It is important to use in the inner loop some appropriate optimization criterion that helps to increase stability.

At first sight it seems compelling to solve the two problems of periodic gait generation and stability optimization simultaneously in a one-level optimization procedure. Theoretically this should result in increased stability as more variables are optimized with respect to

this overall goal. But the cost of that is the need to solve a considerably more difficult optimization problem. And, in contrast to the two-level optimization problem, the iterates before convergence do not represent valid periodic gaits since the constraints are only satisfied in the optimum.

The two-level stability optimization procedure delivered excellent solutions for all walking robot problems investigated in our research. However the author can not rule out the possibility that there exist other types of dynamical systems for which there is no natural split of variables as depicted in figure 1.8 and for which two-level approach does not provide satisfying solutions; hence the one-level approach should be used. We have performed preliminary studies for this one-level procedure and will come back to this issue at different points in this thesis showing possible implications and solution methods.

To summarize, for the two-level optimization procedure described above, the following subtasks needed to be solved:

- choose physical robot models to be studied
- set up a correct description for periodic gaits of those models
- find a periodic solution for a given set of parameters that satisfies some appropriate optimization criterion on the basis of efficient and reliable numerical methods
- define stability criterion and formulate adequate objective function f_{stab}
- develop numerical methods for the optimization of stability by variation of parameters and find solutions for models selected above
- analyze results.

How we have solved these tasks will be presented in the following chapters.

Chapter 2

Modeling Periodic Robot Gaits

The dynamical properties of humans, animals and robots can be represented by sets of highly nonlinear differential and algebraic equations. In the case of walking motions additional difficulties arise due to non-smoothness and multiple phases. The non-smoothness is caused by intermittent ground contacts and by limbs reaching the joint limits which implicitly depend on the configuration variables of the system and usually result in discontinuities of the velocity variables. Very often, gaits involve different phases of motion which have varying degrees of freedom (DOF) and are characterized by different sets of equations. In the case of running motions, e.g., one has to distinguish between alternating flight phases and one-leg-contact phases. Additionally, the nonlinear dynamics have to satisfy periodicity constraints on all or a subset of the position and velocity variables. By models of periodic gaits we understand the entity of all dynamic equations of the multibody-system and of phase change conditions and collision rules, periodicity constraints and all other constraints characterizing a specific gait.

Following the robot types introduced in the previous chapter, our models include passive gaits (resulting in autonomous differential equations) and actuated gaits (non-autonomous equations depending explicitely on time). As inputs for the actuated systems we typically chose joint torques or forces which are a natural choice for robots. To fully capture the open-loop properties of human gaits, models of the muscular activity would have to be included in the dynamic equations. This field has been investigated in the author's diploma thesis (Metzger [57]). As stability is the main focus of this thesis and specific muscular properties do not seem to be a key factor for this we have chosen to stick with basis actuators. Nevertheless, the biped models treated here can be regarded as good descriptions of the main gait features for both robots and humans.

In this chapter we present our general approach to model gaits in biology and robotics. When formulating the models we pursue two principal goals. On one hand, the models have to lead to a realistic-looking natural motion, i.e. they should neither be too edgy nor too springy. On the other hand, they should be suited to act in combination with numerical simulation and optimization methods.

In section 2.1 we introduce the general form of periodic multi-phase gait models. In section 2.2 we explain how to generate model equations starting from a physical model in two or three dimensions. In order not to tear apart things that belong together, gait models for the specific open-loop stable walking robots treated in this thesis are deferred to chapters 7, 8, and 9 also containing the stability optimization results for the respective models. We have implemented all robot models with the components described in this chapter in the framework of an object-oriented modeling library as described in section 2.3.

2.1 Description of Periodic Gaits with Multiple Phases

Gaits should be formulated as multi-phase problems. Modeling gaits by a single set of equations that describe e.g. ground contacts by continuous forces growing exponentially close to the ground would lead to very stiff equations and therefore to small integrator step sizes and long computation times and should thus be avoided (see Simon [84]). A global gait model would always be either too stiff for integration or too smooth to be a good description of reality. A model change should therefore be performed when a foot reaches ground level. The same is true for hard joint contacts, like a kneestrike in walking motions. These model changes lead to what we call a multi-phase problem.

In this section the general form of gait models with multiple phases is presented. We start with the equations of motion for each phase and phase transition conditions in section 2.1.1 and periodicity constraints in section 2.1.2. Different approaches to the formulation of contacts are evaluated in section 2.1.3. In section 2.1.4 we finally argue why the order of phases should be imposed for a multi-phase problem.

2.1.1 Equations of Motion for a Phase

In a multi-phase problem every phase is described by a separate set of equations of motion, the form of which is introduced in this section.

We start with the well known standard form of Newton's law

$$M \cdot a = f \tag{2.1}$$

with mass matrix M, accelerations a, and right hand side force vector f. Showing the dependencies of the terms in equation (2.1) we get

$$M(q(t), v(t), p) \cdot a = f(q(t), v(t), u(t), p) \tag{2.2}$$

where t denotes the physical time, q and v the position and velocity coordinates corresponding to a, p the model parameters, and u the actuations (or, in mathematical terms, control or input functions – which are not to be confused with feedback controls!). In the passive or autonomous case, f does not depend on $u(t)$.

Note that the state of a mechanical system, being a second order system by nature, is not completely defined by position variables alone. Corresponding velocities also need to be indicated, such that we can always write the vector of state variables x as $x^T = (q, v)^T$. With that, equation (2.2) can be rewritten as explicit first-order system of ordinary differential equations (ODE) in x:

$$\dot{q}(t) = v(t) \tag{2.3}$$
$$\dot{v}(t) = a(t) = M^{-1}(q(t), v(t), p) \cdot f(q(t), v(t), u(t), p). \tag{2.4}$$

All the above equations are only valid if q represents a set of independent coordinates for the robot.

In the more general case, one might use redundant coordinates for the description and simulation of complex gait models. This results in a form of multibody system equations different from (2.2), the so-called descriptor form:

$$M(q(t), v(t), p) \cdot a = f(q(t), v(t), u(t), p) - G^T(q(t), p)\lambda \tag{2.5}$$
$$g_{pos}(q(t), p) = 0 \tag{2.6}$$

with the Lagrange multipliers λ, the constraint equations g_{pos}, and their partial derivatives $G = \frac{\partial g_{pos}}{\partial q}$. System (2.5)/(2.6) is a differential algebraic equation (DAE) of index 3. Differentiating the constraints (2.6) twice with respect to time leads to the index 1 system

$$\dot{q}(t) = v(t) \tag{2.7}$$
$$\dot{v}(t) = a(t) \tag{2.8}$$
$$\begin{pmatrix} M(q(t), v(t), p) & G^T(q(t), p) \\ G(q(t), p) & 0 \end{pmatrix} \cdot \begin{pmatrix} a \\ \lambda \end{pmatrix} = \begin{pmatrix} f(q(t), v(t), u(t), p) \\ \gamma(q(t), v(t), p) \end{pmatrix} \tag{2.9}$$

with the differential variables q and v, the algebraic variables a and λ, and the abbreviation

$$\gamma(q(t), v(t), p) = -v^T \frac{d\, G(q(t), p)}{d\, q}\, v. \tag{2.10}$$

Assuming that the mass matrix M is positive definite on the null space of G and that G has full rank, the left hand side matrix in equation (2.9) is regular and the system can be uniquely solved for the accelerations a in each step. These are fed into the right hand side of equation (2.8) such that the system (2.7)/(2.8) can be treated like an ODE. Additionally, the invariants on position and velocity level

$$g_{pos} = g(q(t), p) = 0 \tag{2.11}$$
$$g_{vel} = G(q(t), p) \cdot \dot{q}(t) = 0. \tag{2.12}$$

have to be satisfied.

The individual phases are separated by implicitely defined switching points at which a switching function is zero:

$$s_i(t, q(t), v(t), u(t), p) = 0 \tag{2.13}$$

At those points, discontinuities of the velocities $\Delta v(t, q, v, u, p)$ and/or the right hand side $\Delta_{RHS}(t, q, v, u, p)$ (i. e. discontinuities in the accelerations) can occur.

2.1.2 Periodicity Constraints

For periodic gaits, all velocities and most position variables – except possibly for those describing the direction of travel – are periodic with cycle time T:

$$q_{(red)}(T) = q_{(red)}(0) \tag{2.14}$$
$$v(T) = v(0). \tag{2.15}$$

In the case of bipedal gaits, it often makes sense to consider only one step, and not a full cycle consisting of two steps. Formulation of periodicity constraints on one step (including a reflection of legs) leads to perfectly symmetric results where every left step equals to every right step and eliminates e. g. limping gaits. But if unsymmetric gaits are of practical interest for a specific problem, the full cycle has to be considered.

2.1.3 Models for Ground and Joint Contacts

Gaits are characterized by alternating ground contacts of the feet. The interaction of robot and ground has to be described in the model. There are two fundamentally different ways to model ground contacts, either based on forces or based on constraints:

- Modeling the ground reaction forces in vertical and horizontal directions:
  e.g. describing the frictional forces by a viscous or Coulomb friction model and the normal forces by linear or nonlinear spring-damper elements. This describes an elastic or elasto-plastic impact taking a finite amount of time. The DOFs of the system remain unchanged. In this case, touchdown causes only discontinuities in the right hand side and the accelerations, but none in the velocities.

- Modeling the constraints caused by a rigid and frictional ground:
  i. e. fixing the foot contact point velocity to zero immediately after impact. This results in discontinuities of all system velocity variables. Constraint based model changes usually modify the number of DOFs of the system.

Thorough testing of both types of contact models have shown that the constraint-based modeling approach is to be preferred as it leads to more natural motions.

If a joint is flexed or extended to its limits, similar considerations apply.

If a constraint-based contact model is used, velocity changes caused by this inelastic impact will have to be computed. Equations describing conservation of momentum over the collision are used for this purpose. For ground contact, we work on two basic assumptions:

- Lift-off is impulse-free, i.e. there are no discontinuities in the velocity. This is applied to a lift-off of the swing leg in a walking motion as well as to the lift-off of the whole robot during a running or hopping motion.

- Touchdown is generally associated with an impulse and therefore velocity discontinuities might occur depending on the collision model chosen (see above).

Similar assumptions hold for start and end points of joint contacts.

2.1.4 Prescribing the Order of Phases

We have seen above that constraint-based contact models are desirable since they are most accurate from a physical point of view. But to be able to use such methods, the order of phases of the motion has to be a priori prescribed.

Leaving the phase order free to be determined by continuous optimization methods would restrict us to smooth phase transitions for numerical reasons: Only with transition between any two succeeding model phases being twice continuously differentiable and the number of DOFs being the same for all phases we would be able to compute meaningful derivatives of the integration end values with respect to initial values. In general, this approach does not lead to very natural walking motions.

Our numerical method allows us to impose only the order of phases but to leave all individual phase times free.

One alternative approach is to formulate the problem with free phase sequence as mixed-integer optimization problem. But for walking mechanisms with few legs not many possible meaningful orders to choose from exist: For a biped, we can basically distinguish between walking and running, and even for a quadruped there are only a very few possibilities, namely pace, trot, and different sorts of gallop. It is therefore possible to solve the mixed-integer problem by enumeration and perform individual optimization runs for each possible order of phases.

2.2 Deriving the Equations of Motion

After we have seen the general forms of equations of motion in the last section, we will explain here how to set up equations (2.3) - (2.3) or (2.7) - (2.9) for given physical robot models. It is not the intent of the author to provide a complete overview of all possible methods to generate equations of motion - these can be taken from mechanics textbooks like Greenwood [35] and Hauger et al [38], or university manuscripts like Eppler [27]. In this section we only explain the solution approaches we have chosen to take, both for the selection of coordinates and the establishment of the equations of motion. The reasons for the specific choices are given. We generally derive the equations of motion and collision rules for our robots using symbolic mathematics packages like Maple® .

2.2.1 Coordinate Choices

For simulation and optimization, physically meaningful coordinates should be chosen as this simplifies the generation of initial values for simulation and of initial trajectories for the optimization. It also allows for a more comfortable interpretation of results. From an engineering point of view, intuitive coordinate choices for walking robots are orientation angles of bodies (e. g. Euler angles) or position vectors of predominant points, e. g. the hip. We have seen before that phases of motion usually have different numbers of DOF. We nevertheless like to choose the same set of coordinates for all phases. In general we choose minimal coordinates for the phase(s) with most DOFs, that are redundant coordinates for the other phase(s), such that we obtain ODE and DAE descriptions, respectively.

One peculiarity of walking motions for which at every instant at least one foot is in contact with the ground is that a set of coordinates adequate to describe one step and thus completely specify the periodic motion does not allow to directly access the full configuration space of a multi-step walk. This is due to the piecewise holonomic, but overall non-holonomic nature of walking motions. For running and hopping motions that include phases without ground contact, this effect does not occur.

When deriving the equations of motion several other auxiliary coordinates naturally occur which do not belong to the chosen set of coordinates for simulation and optimization but which are needed to express the system's dynamics. Examples for such coordinates are position and orientation variables of the individual bodies, or derivatives of these. As those coordinates depend on the chosen method for the generation of equations, the corresponding implications will be explained when we discuss specific methods in the next sections and when presenting our robot models later on. These auxiliary coordinates can be symbolically eliminated during the model setup process in order to obtain analytic expressions in the optimization coordinates for all entries of the equations of motion. For complex systems this leads to extremely involved expressions filling many lines of code to express one single mass matrix or right hand side entry. For most of our models, we have therefore chosen to keep the additional equations and coordinates and numerically solve in every integration step the resulting larger system using a regular or a sparse solver. This gives us, among other quantities, the accelerations corresponding to positions and velocities in our set of optimization variables.

This hybrid coordinate choice has the following advantages:

- reduced error proneness as no symbolic operations are necessary,
- increased flexibility for model changes, e. g. joint modifications or the addition of bodies
- increased ability to monitor the auxiliary variables, which often are physically interesting quantities.

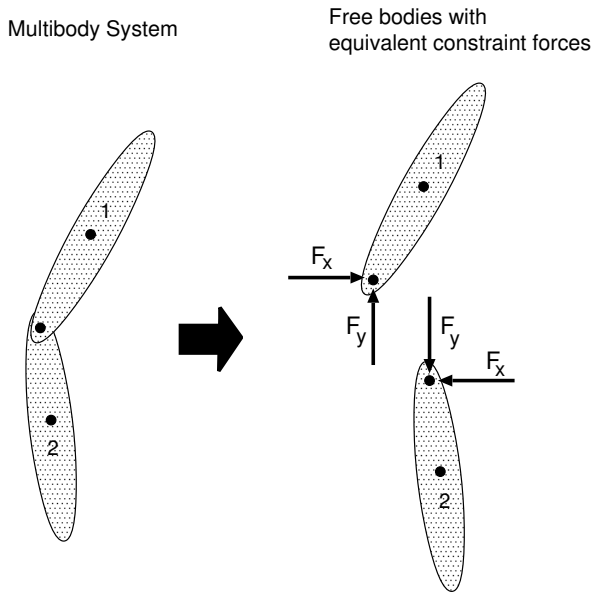


Figure 2.1: Generating the free body diagram for a simple multibody system

2.2.2 Deriving Two-dimensional Equations of Motion by Free Body Diagrams

We have used free body diagrams to set up two-dimensional equations. It is a very intuitive method if applied to planar motion and is based on a synthesis of information gathered from all bodies. For a detailed description, see e. g. Dinkler [25]. As demonstrated in figure 2.1, all joints and connections between the individual bodies of a multibody system are removed and replaced by equivalent constraint forces and/or torques. External contacts e.g. between body and ground are also characterized by constraint forces.

For each isolated body (index i) with mass m_i and moment of inertia $\Theta_{i,z}$ Newton's laws of motion in translational and rotational directions can easily be written down in the form:

$$m_i \ddot{r}_{i,x} = F_{i,x} \tag{2.16}$$
$$m_i \ddot{r}_{i,y} = F_{i,y} \tag{2.17}$$
$$\Theta_{i,z} \ddot{r}_{i,\phi} = M_{i,z} \tag{2.18}$$

F_i and M_i are the sums of all external and constraint forces/torques acting on body i. For a two-dimensional system with n bodies we obtain $3n$ dynamical equations. Additionally, kinematic equations describing the relation between position variables r_x, r_y and orientation variables r_ϕ (and their first and second derivatives) of neighboring bodies i and j introduced by the joints and contacts have to be established:

$$k_{i,j}(r_{i,x}, r_{i,y}, r_{i,\phi}, \dot{r}_{i,x}, \dot{r}_{i,y}, \dot{r}_{i,\phi}, \ddot{r}_{i,x}, \ddot{r}_{i,y}, \ddot{r}_{i,\phi}, r_{j,x}, r_{j,y}, r_{j,\phi}, \dot{r}_{j,x}, \dot{r}_{j,y}, \dot{r}_{j,\phi}, \ddot{r}_{j,x}, \ddot{r}_{j,y}, \ddot{r}_{j,\phi},) = 0 \tag{2.19}$$

The number of kinematic equations is at least $3n - n_d$ (with n_d being the number of DOFs), possibly more if additional auxiliary coordinates are used. The resulting system of all dynamic and kinematic equations is linear in the second derivatives of the position and orientation coordinates of each body and in the constraint forces. Those are the model setup coordinates we have discussed in the previous section.

We prefer this synthetic method to the analytic approach of using Lagrange's equations, which is another very popular method in mechanics, compare e. g. Greenwood [35]. Lagrange's equations of the second kind for a set of n_d independent generalized coordinates q_i can be written as

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i \qquad i = 1, 2, ..., n_d \tag{2.20}$$

where L is the Lagrangian function

$$L = T - V. \tag{2.21}$$

V and T are the total potential and kinetic energy of the system, and Q_i are the generalized forces in the direction of q_i not derivable from a potential function (e.g. frictional forces, non-holonomic constraint forces etc.).
For redundant generalized coordinates q_i (dimension n_r), Lagrange's equations of the first kind have to be applied:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i - G_i(q)^T\lambda \qquad i = 1, 2, ..., n_r \tag{2.22}$$

with the additional constraint force term on the right hand side.

Why do we not use Lagrange's equations? In both cases, Lagrange's equations force us to use the same set of coordinates for optimization and for the model setup. T and V always need to be expressed in terms of the generalized coordinates which again leads us to the involved equations we wanted to avoid!

2.2.3 Deriving Three-dimensional Equations of Motion by Angular Momentum Balances

For three-dimensional systems we have chosen the - in this case - more intuitive approach to derive equations based on overall balances of translational and rotational momentum. In three dimensional space, the method of free body diagrams reaches its limits of application as it becomes considerably harder to correctly describe the directions of constraint forces and accelerations.

For three-dimensional systems it seems more intuitive to derive the equations of motion based on overall balances of translational and rotational momentum

$$F = m\ddot{r}_C = \sum_i^n m_i\ddot{r}_i \tag{2.23}$$

$$M = \dot{H}. \tag{2.24}$$

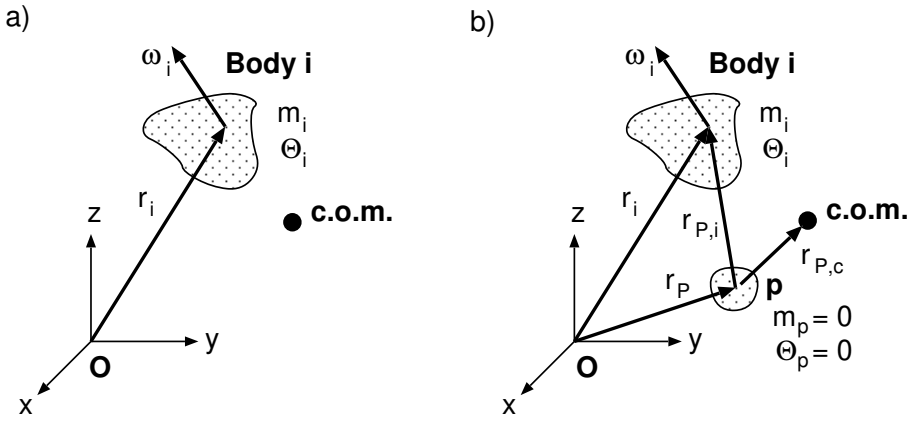


Figure 2.2: Derivation of total angular momentum of a multibody system a) about origin b) about an arbitrary (moving) point

F and M are the sums of all external forces and torques, the index C denotes the center of mass of the whole system, m is the total mass, and H is the total angular momentum. The reference point for H is either the center of mass or a fixed point. We will see later how equation (2.24) needs to be modified for general reference points. Using the center of mass as reference point has the advantage that translational and rotational motion can be treated independently. On the other hand, the center of mass of a multibody system is no physical point fixed to any of the bodies but an imaginary point constantly moving relative to all of them. Frequently, certain points in the multibody system are inertially fixed and the overall motion can be considered as a rotation about this fixed point.

In the following, we will concentrate on deriving the rotational equations about general reference points that are not necessarily the center of mass or a fixed point.

The total angular momentum of a multibody system about the origin (see figure 2.2a) is found by summing the angular momenta of the individual bodies.

$$H = \sum_{i=1}^{n} (r_i \times m_i \dot{r}_i) + \sum_{i=1}^{n} (\Theta_i \omega_i) \tag{2.25}$$

where m_i are the masses, Θ_i the moments of inertia, r_i the center of mass positions, and ω_i the angular velocities of the n bodies. r_i and ω_i are described in global coordinates. The moments of inertia Θ_i are described in body fixed coordinates. Due to the rotational motion of the bodies their derivatives in the inertially fixed coordinate system are not zero.

With this, the derivative of the total angular momentum becomes

$$\dot{H} = \sum_{i=1}^{n} (r_i \times m_i \ddot{r}_i) + \sum_{i=1}^{n} (\Theta_i \dot{\omega}_i + \omega_i \times \Theta_i \omega_i) \tag{2.26}$$

as

$$\dot{r}_i \times m_i \dot{r}_i = 0. \tag{2.27}$$

Thus, we obtain for the overall rotational equation about the origin:

$$\sum_{i=1}^{n}(r_i \times m_i\ddot{r}_i) + \sum_{i=1}^{n}(\Theta_i\dot{\omega}_i + \omega_i \times \Theta_i\omega_i) = M \tag{2.28}$$

In the more general case, we choose a reference point P located on an additional body of mass 0 (figure 2.2b). P may perform arbitrary translational motions (and rotational motions, which however do not play a role as the body is massless). The total angular momentum of the system about P is

$$H = \sum_{i=1}^{n}(r_{P,i} \times m_i\dot{r}_{P,i}) + \sum_{i=1}^{n}(\Theta_i\omega_i). \tag{2.29}$$

For a general reference point, equation (2.24) needs to be modified:

$$M_P - r_{P,C} \times m\ddot{r}_P = \dot{H}_P. \tag{2.30}$$

The additional term $-r_{P,C} \times m\ddot{r}_P$ can be interpreted as moment about P produced by the inertia force $m\ddot{r}_P$ that results from the translational motion of the reference frame. The term is zero for any reference point that is inertially fixed or moving at constant speed.

The rotational equation about a general accelerated reference point P becomes

$$\sum_{i=1}^{n}(r_{P,i} \times m_i\ddot{r}_{P,i}) + \sum_{i=1}^{n}(\Theta_i\dot{\omega}_i + \omega_i \times \Theta_i\omega_i) + r_{P,C} \times m\ddot{r}_P = M. \tag{2.31}$$

With

$$r_{P,C} \times m\ddot{r}_P = \sum_{i=1}^{n} r_{P,i} \times m_i\ddot{r}_P, \tag{2.32}$$

we can also write

$$\sum_{i=1}^{n}(r_{P,i} \times m_i\ddot{r}_i) + \sum_{i=1}^{n}(\Theta_i\dot{\omega}_i + \omega_i \times \Theta_i\omega_i) = M. \tag{2.33}$$

It goes without saying that these equations can also be applied to parts of the multibody system. Following the principle of free body diagrams, possible constraint forces and torques caused by the rest of the system have in this case to be taken into account because they become 'external' from this point of view.

2.3 Numerical Implementation of Gait Models

We have implemented all our robot models in the framework of an object oriented model library in C++. Every model class encapsulates all information about a model including basic model data like dimensions and all features previously defined in this chapter. This implementation has the advantage of ensuring uniform interfaces to all mechanical models. No model modifications are necessary for use with other software, e.g. integrators. The following listing characterizes the Model data type:

```
class Model{

 protected:
  int itsNoOfStates;
  int itsNoOfControls;
  int itsNoOfParams;
  int itsNoOfModels;
  int itsModelNo;
  int itsNoOfSwitches;
  double itsTime;
  Vector<double> ItsX;
  Vector<double> ItsU;
  Vector<double> ItsP;

 public:
  Model();
  Model(Model& OtherModel);
  virtual ~Model();
  void update(double time, Vector<double>& X, Vector<double>& U,
      Vector<double>& P, int modelNo = 1);
  virtual void calcRHS(Vector<double>& XDot) = 0;
  virtual void calcJump(int switchNo, Vector<double>& XAfterJump) = 0;
  virtual double calcSwitchingFunction(int switchNo) = 0;
  virtual double confirmSwitch(int switchNo) = 0;
  virtual void calcSwitchingDerivs(int switchNo, Vector<double>& DswDx,
      Vector<double>& DswDu, Vector<double>& DswDp, double& dswDt) = 0;
  int getNoOfStates();
  int getNoOfControls();
  int getNoOfParams();
  int getNoOfSwitches();
  virtual void print(int printLevel);
  virtual void print(int printLevel, ofstream& fout);

};
```

The equations of motion of the robot are internally established based e.g. on free body diagrams and Newton's laws or conservation of momentum balances. As described in section 2.2.1, typically larger sets of coordinates than those for simulation and optimization are used for this purpose. These systems of equations are internally solved, and only optimization coordinates are visible to the outside. The right hand side for the equations of motion is exported on request. Switching function values and jump discontinuities at switching points, that result from complicated momentum balances are also internally computed on demand.

Chapter 3

Mathematical Methods for the Generation of Optimal Periodic Gaits

The task of gait generation determines, for a given mechanical model and a given set of model parameters, the initial values for the position and the velocity variables, the cycle time, and – for actuated systems – the actuator inputs such that the periodicity conditions and other restrictions are fulfilled. The task also includes a detection of non-feasibility for a given configuration.

The question for optimal periodic gaits can arise in different circumstances.

In this thesis, we are especially interested in the search for periodic gaits as subtask in the inner loop of the two-level optimization procedure, as outlined at the end of chapter 1. An optimization criterion has to be added in this case to make the solution of this subproblem unique and to support the objectives of the outer loop.

But optimal periodic gaits are also of importance as stand-alone problems. Forward simulation of gait models without any information about suitable actuator histories is very unlikely to result in physically meaningful trajectories. Optimization methods can help to determine actuator inputs creating periodic gaits. And frequently the question for an optimal motion, like the fastest, slowest, most energy efficient etc. gait arises.

The one-level stability optimization problem mentioned in chapter 1 does of course also ask for the generation of an optimal periodic gait, but it does not fall into the range of problems treated in this chapter. Both of the above problems of optimal gait generation – stand-alone and inner loop task – are complex, but standard optimal control problems. Their objective functions are of standard separable Mayer or Lagrange type. We will see in chapter 5 that the objective function characterizing stability does clearly not belong into this category such that the methods described in this chapter are only partly applicable.

While there is no tool available to solve such a non-standard optimal control problem,

software for the solution of standard optimal control problems already exists today, in particular the powerful code MUSCOD, that has been developed in the research group of Bock & Schlöder by Leineweber [48], [47] on the basis of the original code of Bock & Plitt [11] for stiff chemical systems of ODE or index 1. MUSCOD relies on fast and efficient integrator libraries that are also capable to compute derivatives of the trajectories, like ODEOPT that has been incorporated into the optimal control code by its author Winckler [102]. In this study, we use a variant of MUSCOD for the generation of optimal periodic gaits. It was coupled with the gait model library described in the previous chapter.

The purpose of this chapter is to introduce theoretical background, problem formulation and solution methods for optimal periodic gaits in the sense of standard periodic optimal control problems.

As differential equations and dynamical systems are closely related fields of research, in section 3.1 we take the dynamical systems point of view and recall some important terminology from that area. In section 3.2 we introduce some theory about existence and uniqueness of solutions of (periodic) boundary value problems. From this we conclude in section 3.3 that the formulation of gait problems should always lead to optimal control problems and not only to boundary value problems. We give the full formulation of a standard multi-phase optimal control problem with discontinuities. Section 3.4 is finally dedicated to the numerical solution of periodic optimal control problems by the direct multiple shooting approach.

3.1 The Dynamical Systems Point of View

A system, the behavior of which is described by differential equations — like a walking robot – is also called a *(continuous-time) dynamical system.* In this section we introduce some basic terminology common in the field of dynamical systems because it is often used in the context of walking robots and sometimes differs from the terminology customary in the field of optimization and simulation. As references for this section we have used the books of Kuznetsov [44] and Baker & Gollub [5] and the relevant chapter in Bronstein et al.[13].

The *state space* X of all dynamic variables of the system is equivalently called *phase space.*

The *evolution operator* of a dynamical system describes the deterministic process of evolution in time t. It is the map ϕ^t that transforms an initial state $x_0 \in X$ into some state $x_t \in X$ at time t: $x_t = \phi^t x_0$. In the continuous-time case, the family $\phi^t{}_{t \in I_T}$ of evolution operators is called a *flow.*

An *equilibrium* or *fixed point* x_0 is a point in state space that is mapped onto itself by the evolution operator for arbitrary times t:

$$\phi^t x_0 = x_0 \text{ for all } t \in I_T. \tag{3.1}$$

From this follows that a system put into equilibrium state remains there forever.

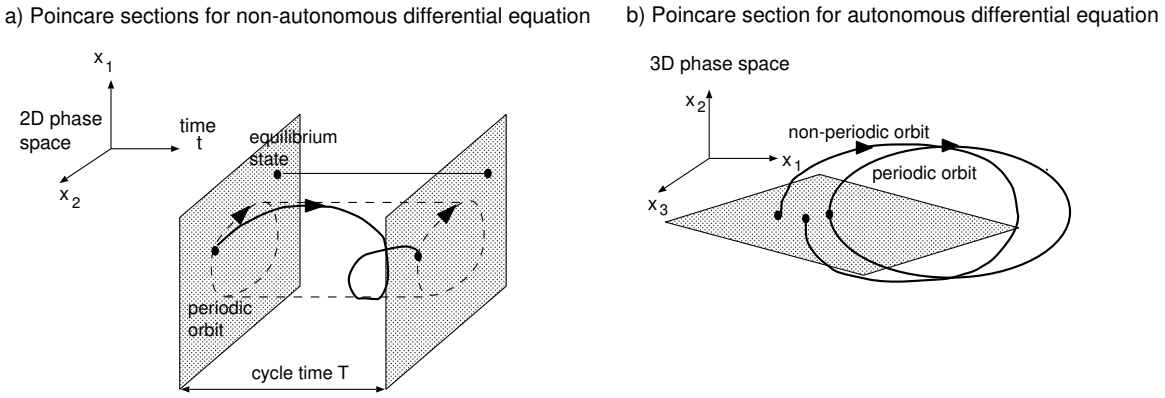


Figure 3.1: Definition of Poincaré sections for a) non-autonomous and b) autonomous differential equations

Orbits of a continuous-time system with a continuous evolution operator are curves in the state space X parameterized by the time t and oriented by its direction of increase. Orbit is therefore a synonym for trajectory.

A *periodic orbit* or *cycle* L_0 is an orbit for which each point $x_0 \in L_0$ satisfies

$$\phi^{t+T} x_0 = \phi^t x_0 \text{ with some } T > 0, \text{ for all } t \in I_T. \tag{3.2}$$

The minimal T with this property is called the *period* of the cycle L_0. In the continuous-time case, a cycle L_0 is represented by a closed curve. If a cycle is isolated, i.e. there are no other cycles in its neighborhood, it is called a *limit cycle*.

Phase portraits are diagrams containing a visualization of possible orbits in phase space. In practice it is of course impossible to show all orbits, so only several key orbits are depicted.

A very helpful tool for the study of periodic dynamical systems are *Poincaré sections* which can be defined as 'snapshots' of the system's motion taken at regular intervals T in time (typically the cycle time). In the case of autonomous systems it is customary to define the Poincaré section as $(n-1)$-dimensional manifold in n-dimensional phase space that is orthogonal to the orbit. The results are the same if applied to a T-periodic orbit but a distinction becomes obvious for a (non-periodic) perturbation of this orbit for which the time between two intersections with this manifold is not equal to T. Poincaré sections for non-autonomous and autonomous systems are shown in figure 3.1.

The map that transforms a state on one Poincaré section to a corresponding state on the next section is called the *Poincaré map* associated with the T-periodic cycle L_0. The computation of Poincaré maps and their Jacobians for both autonomous and non-autonomous systems will be a core component of our stability optimization procedure to be described in later chapters.

3.2 Existence and Uniqueness of Solutions of Boundary Value Problems

Gait models as described in chapter 2 result in periodic multi-phase boundary value problems with discontinuities – so far without any optimization criterion. In this section, we will briefly review some important facts about existence and uniqueness of solutions of boundary value problems. In contrast to initial value problems for which existence and uniqueness can generally be guaranteed under mild assumptions of continuity and Lipschitz continuity, for boundary value problems non-uniqueness or non-existence can occur even for very simple cases. The material of this section is treated more extensively in Walter [98], Werner & Arndt [100], and Ascher et al. [4].

The basic form of a boundary value problem for first order systems of order n is

$$\begin{aligned} \dot{x} &= f(t, x(t)), \quad t \in [a, b] && (3.3) \\ r(x(a), x(b)) &= 0 && (3.4) \end{aligned}$$

In contrast to initial value problems, boundary conditions are specified at two different points a and b. The formulation of a boundary value problem with boundary conditions of type (3.4) is obviously only possible for a first order system with at least two differential equations.

In the case of a periodic boundary value problem, boundary conditions (3.4) take the form

$$r(x(a), x(b)) = x(b) - x(a) = 0. \tag{3.5}$$

In analogy to the classical two-point boundary value problem (3.3)/ (3.4) we can also define a multi-point boundary value problem with boundary conditions specified at p different points

$$r(x(t_1), x(t_2), ..., x(t_p)) = 0. \tag{3.6}$$

A boundary value problem is called linear if its differential equations as well as its boundary conditions are linear. The following theorem is taken from Werner & Arndt [100].

Theorem 3.1 *The linear boundary value problem*

$$\begin{aligned} \dot{x}(t) + A(t)x(t) &= c && (3.7) \\ Mx(a) + Nx(b) &= d && (3.8) \end{aligned}$$

has a unique solution for arbitrary c, d if for a fundamental matrix $X(t)$ of the homogenous system $\dot{x}(t) + A(t)x(t) = 0$ we have:

$$\Delta := MX(a) + NX(b) \tag{3.9}$$

is non-singular. If it is singular, the homogenous boundary value problem (with $c = d = 0$) has a number of non-trivial solutions.

For boundary value problems involving nonlinear differential equations there is no way to determine existence nor uniqueness of solutions for the general case. Boundary value problems are 'problems in the large', and existence of solutions of differential equations can often only locally be guaranteed.

In many cases of practical interest, however, there is a number of possible solutions. Some useful theoretical results for the general case can be derived expressing boundary value problem (3.3)/ (3.4) in terms of an associated initial value problem

$$\dot{y} = f(t, y(t)), \quad t > a \tag{3.10}$$
$$y(a) = s \tag{3.11}$$

where s is an n-dimensional parameter vector. Assuming Lipschitz continuity, there is a unique solution $y(t; s)$ of the initial value problem. $x(t) = y(t; s)$ for each s that solves the boundary value problem if s is chosen such that

$$g(s) = r(s, x(b; s)) = 0 \tag{3.12}$$

This is a system of n nonlinear equations for n unknowns s_i that may have one solution, many or none at all. See Ascher et al. [4] for the following theorem:

Theorem 3.2 *If $f(t, x(t))$ is continuous for $t \in [a, b]$ and arbitrary x and satisfies a uniform Lipschitz condition, then the nonlinear boundary value problem (3.3)/ (3.4) has as many solutions as there are distinct roots in equation (3.12).*

It remains to draw conclusions for the problem of gait generation. A typical straightforward boundary value problem arising in this context of non-actuated systems would be to determine a periodic solution for a given set of model parameters and free initial values of states and cycle time. This can be written as an extension of (3.3)/(3.5):

$$\dot{x} = f(t, x(t, p), p), \quad p \in \mathbb{R}^k, \ t \in [a, p_{k+1}] \tag{3.13}$$
$$0 = x(a, p) - x(p_{k+1}, p) \tag{3.14}$$

Such a system may have a unique solution, multiple solutions or no solution at all. Very often, the system has additional possibilities of variation, like controls, and also additional restrictions, like constraints on states, parameters and controls. The former even increases the chances of non-uniqueness, the latter those of non-solvability. But locally non-unique solutions may lead to a failure of the numerical method.

We may therefore conclude that formulating a gait problem as simple boundary value problem may be the source of numerical problems and should thus be avoided. Measures to make the solution unique and for a detection of non-solvability have to be taken.

3.3 Formulation of Periodic Gaits as Multi-phase Optimal Control Problem with Discontinuities

A solution to the non-uniqueness problems resulting from a boundary value problem formulation outlined in the last section is to add some appropriate optimization criterion.

This leads to the formulation of a multi-phase optimal control problem

$$\min_{x,u,T} \quad \int_0^T \phi(x(t),u(t),p)\,dt \quad + \quad \Phi(T,x(T),p) \tag{3.15}$$

$$\begin{aligned}
\text{s. t.} \qquad \dot{x}(t) &= f_j(t,x(t),u(t),p) \quad \text{for} \quad t \in [\tau_{j-1},\tau_j], \\
&\qquad\qquad j=1,...,n_{ph},\ \tau_0=0, \tau_{n_{ph}}=T && (3.16)\\
x(\tau_j^+) &= h(x(\tau_j^-)) \quad \text{for} \quad j=1,...,n_{ph} && (3.17)\\
g_j(t,x(t),u(t),p) &\geq 0 \qquad \text{for} \quad t \in [\tau_{j-1},\tau_j] && (3.18)\\
r_{eq}(x(0),..,x(T),p) &= 0 && (3.19)\\
r_{ineq}(x(0),..,x(T),p) &\geq 0. && (3.20)
\end{aligned}$$

Equations (3.16) – (3.20) alone would represent a multi-point boundary value problem with additional nonlinear inequality constraints.

The state variables x combine the position and velocity variables q and v of the mechanical system. p are the model parameters, and u the control variables or actuations. This formulation covers actuated as well as passive systems; for the latter the dimension of u simply is zero. In the context of the two-level stability optimization procedure the model parameters p are fixed in this subproblem and are only varied in the outer stability optimization loop (see section 1.5). For a stand-alone problem of periodic gait generation it could make sense to also allow for a model parameter variation in this optimal control context, and equation (3.15) would have to be accordingly modified.

For simplicity, we have represented here only the case of ODE models. If the motion is characterized by DAEs, equations (3.16) have to be replaced by the corresponding formulation of type (2.5)/(2.6). The full cycle $[0,T]$ is divided into phase intervals $[\tau_j, \tau_{j+1}]$ with possibly different dynamical models (3.16) and different continuous inequality constraints on states and controls (3.18).

State and right hand side discontinuities are only allowed between phases. Switching function handling is not performed during integration. Instead, phase switching at implicitly defined times is monitored by a time transformation (see section 3.4.3) and reformulation of the switching conditions as interior point constraints of type (3.19) (see Bock [9]).

In the sense of the above optimal control problem formulation, periodicity conditions are coupled pointwise equality constraints of type (3.19). Box constraints and more complex continuous constraints (such as foot clearance) on states and controls are included in equation (3.18).

Note that the optimal control problem formulation in equation (3.15) allows for objective functions of Lagrange type as well as of Mayer type involving trajectory end values. Those two types are equivalent from a mathematical point of view and can be transformed into each other.

For the optimization of walking motion there are several classical objective functions of interest.

Energy consumption is an important issue although there are different opinions on how to measure energy. It can be either judged in terms of mechanical energy as integral of a force along a path or, as is more common in robotics, in terms of actuator torques u (squared to eliminate direction)

$$\int_0^T ||u||_2^2 dt \tag{3.21}$$

which depend on the electrical energy consumed by the motors. For human walking, a measure of muscular energy consumption would have to be established.

The improvement of performance of walking, running and hopping motions in the sense of speed, height and width is often sought for, and this also leads to objective functions of form (3.15).

For the stability optimization subtask of generating a periodic gait in the inner loop we need to choose an objective function that assists the outer loop stabilization goal rather than working against it. Experience shows that e.g. a maximization of speed would tend to make gaits more unstable instead of more stable. In this sense the following objective functions have proven to be suitable:

- minimization of 'energy input' in terms of torques squared (for actuated robots only)
- minimization of cycle time
- minimization of some characteristic speed.

3.4 Numerical Solution of Periodic Mechanical Multi-phase Optimal Control Problems with Discontinuities

There are some fundamentally different ways to solve periodic optimal control problems.

The approach due to Colonius [21] is very common, e.g. in chemical engineering, where for many systems a steady state solution, corresponding to a static control is known. The basic question then is if it is possible to determine a non-trivial T-periodic control which further improves the steady state value of the objective function. It is answered on the basis of a so-called Π-Test (also see Hartel [37] for a numerical implementation). However,

this approach is not a possible one for the generation of optimal periodic gaits since there are no related steady state solutions in the general case. We are not aware of any case where a constant non-zero control produces a solution to the problem. And for constant zero controls, i.e. performing a transition from actuated to passive walking (if possible at all) – some properties of the system, like the slope angle, would have to be modified. The steady state on flat ground with no controls related to walking motions equals standing still. This being of fundamentally different nature would not help to find a solution for walking.

Among the different possible methods that directly tackle the non-steady periodic optimal control problem we have chosen a direct method based on multiple shooting as it is implemented in MUSCOD (see Leineweber [47]).

In this section we describe the individual steps in the solution process: control discretization, treatment of dynamics, solution of the underlying nonlinear programming problem. We also highlight special aspects of the problem like the optimization of phase times, a possible shift of periodic orbits, and the use of index 3 models in optimal control.

3.4.1 Discretization of Optimal Control Problems

Problem (3.15) - (3.20) represents an infinite-dimensional optimization problem. This section describes how it is transformed into a finite-dimensional optimization problem by discretization of the control functions as well as the system's dynamics and presents the resulting discretized system.

Control Discretization

The classical, so called indirect, approach to optimal control problems is based on Pontryagin's maximum principle and optimality conditions for the infinite problem itself (see e.g. Föllinger [29]). Controls are eliminated by expressing them as functions of state variables and adjoint variables. In theory, the indirect approach has the advantage that solutions with control functions in very general spaces can be characterized but in practice it is very hard to solve the resulting equations even for quite simple cases.

For complex practical problems, generally the direct approach that is based on a discretization of control functions, is favored. Instead of infinite dimensional control functions $u(t)$ as in the indirect approach, parameterizable control functions are used. Control histories of the n_u physical controls are thus described by a finite number of control parameters q_{ij}. One of the special features introduced by Bock & Plitt [11] is to use control functions that have only local support, like piecewise constant or linear functions on a control grid $[t_0, t_1, ..., t_m]$ with m intervals. This has the advantage of leading to a structured optimization problem.

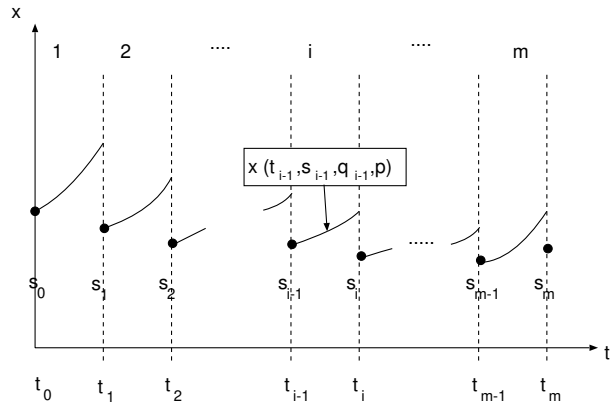


Figure 3.2: Multiple Shooting state discretization

State Discretization

The goal of state discretization is to separate the two tasks of integrating the system's dynamics and solving the optimization problem. For optimal control problems basically the same discretization methods are applicable as for the underlying multi-point boundary value problem.

Resulting from the fact that there is a close theoretical relationship between boundary value problems and initial value problems (see section 3.2) some numerical methods for boundary value problems trace back their solution to that of initial value problems (Ascher et al. [4], Schwarz [82]).

The simplest method of this type is the single shooting method. It treats the full integration interval of the boundary value problem at once. The initial values of the integration are iteratively varied until the boundary conditions of the original boundary value problem are satisfied. A weakness of single shooting is that it is hard or even impossible to find a solution if the initial value problem solution is very sensitive to variations of the initial values.

As the impact of such a high sensitivity is reduced on smaller integration intervals, we are naturally led to the idea of multiple shooting. It splits the long integration interval $[0, T]$ into many smaller ones and introduces the values of the n_x state variables x at all those grid points as new variables s_{ij}. With this approach the original boundary value problem is transformed into a set of initial value problems. Corresponding continuity conditions between integration intervals are added. As proposed by Bock & Plitt [11], it is advantageous for the structure of the resulting problem to choose the same grid for multiple shooting as for control discretization or a subset thereof.
The multiple-shooting approach is especially suited for the generation of periodic gaits for a number of reasons. It allows to impose the desired phase order and switching structure, as the model for each multiple shooting interval can be prescribed differently. The knowledge that one usually has about the trajectory can be used as initial guesses for

the initial values of the states at the multiple shooting points. Gait models often exhibit the above described sensitivity to initial value variations, and therefore multiple shooting increases significantly the chances of finding a solution of the initial value problem and of obtaining sufficiently accurate derivatives.

Discretized Optimal Control Problem

Two more steps need to be taken in the discretization process. First, as the durations of all phases are to be determined by the optimization, they need to be introduced as optimization variables. The necessary transformations are described in section 3.4.3. Second, for all discontinuous physical phase transitions additional artificial phases of duration zero are introduced. These phases formally consist of a single multiple shooting interval.

The discretized optimal control problem becomes

$$\min_{y} \quad \tilde{\Phi}(y,p) \tag{3.22}$$

$$\text{s. t.} \quad \tilde{r}_{con} = x(t_{i+1}, s_i, q_i, p, h) - s_{i+1} = 0 \quad \text{for } i = 0, ..., m-1 \tag{3.23}$$

$$\tilde{r}_{eq,i}(y,p) = 0 \quad \text{for } i = 0, ..., m \tag{3.24}$$

$$\tilde{r}_{ineq,i}(y,p) \geq 0 \quad \text{for } i = 0, ..., m \tag{3.25}$$

with the variable vector $y^T = (s_0, q_0, s_1, q_1, ..., s_m, h)^T$ containing the discretized state and control vectors s_i and q_i at all multiple shooting points/intervals and the vector of phase times h. With dimensions n_x and n_u of the physical state and control vectors, n_{ph} phases, and m multiple shooting/control intervals, the vector y has the dimension

$$n_y = n_{ph} + n_x \cdot (m+1) + n_u \cdot m \cdot k_u, \quad k_u = 1, 2, ..$$

(where the factor k_u depends on the chosen control discretization).

In MUSCOD, constraints can be imposed on all multiple shooting points. Equation (3.24) represents equation (3.19), equation (3.25) combines equations (3.20) and a discretization of (3.18) of the original problem. (3.23) describes the continuity conditions between integration intervals.

3.4.2 Treatment of the System's Dynamics

Please note that the dynamical model is not part of the optimal control problem any more after discretization. Integration has been separated from optimization: the optimization only needs the final values $x(t_{i+1}, s_i, q_i, p)$ of the integration on each interval and the corresponding derivatives calculated by a logically external integrator.

As integration and corresponding sensitivity generation represent a large part of the computational effort necessary to solve the optimal control problem, the use of efficient integrators is crucial. Incorporated into MUSCOD are ODE-integrators of Runge-Kutta type

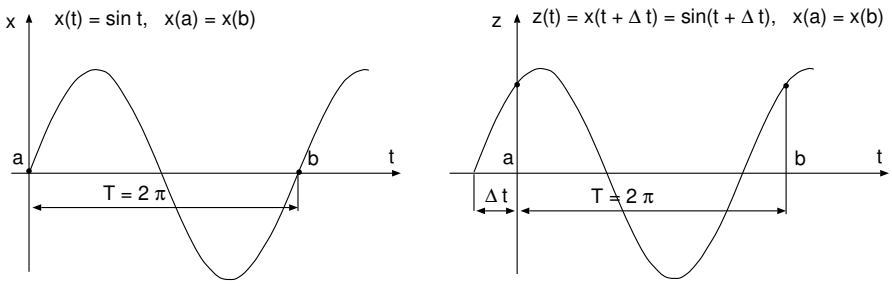


Figure 3.3: Illustration of the shift problem

and ODESIM/ODEOPT of Winckler [102] and the index-1 DAE integrator DAESOL (Bauer et al. [6]). For sensitivity generation, they all rely on the principle of IND (internal numerical differentiation, see Bock [10]) which uses the same discretization schemes for integration and derivative generation and exhibits high numerical stability.

3.4.3 Optimization of Phase Times

As stated above, we want the durations of all phases $h = (h_1, ..., h_{n_ph})^T$ to be free variables of the optimization problem. For this, derivatives with respect to phase times have to be determined during integration. In the context of IND this should not be done using straightforward integration in physical time:

$$x(t_{e,i}) = x(t_{s,i}) + \int_{t_{s,i}}^{t_{e,i}} f(t, x(t))dt. \tag{3.26}$$

By means of a time transformation to the unity interval $t \in [t_{s,i}, t_{e,i}] \rightarrow \tau \in [0, 1]$

$$x(t_{e,i}) = x(t_{s,i}) + (t_{e,i} - t_{s,i}) \cdot \int_0^1 f(t_{s,i} + (t_{e,i} - t_{s,i})\tau, x(t_{s,i} + (t_{e,i} - t_{s,i})\tau))d\tau \tag{3.27}$$

with $t_{s,i} = \sum_{j=1}^{i-1} h_j$, $t_{e,i} = \sum_{j=1}^{i} h_j$ the derivatives with respect to h_i can be computed like derivatives with respect to model parameters.

3.4.4 Handling of the Shift Problem

A classical difficulty occurring in the context of periodic optimal control problems or boundary value problems is the shift problem. Figure 3.3 illustrates this problem for the simple example of the sin-function. It arises if none of the variables at the boundary is fixed to a specific value and only pure periodicity constraints are imposed. Then it follows that if $x(t)$ is a solution of the boundary value problem, any other shifted function $z(t) = x(t + \Delta t)$ for arbitrary Δt is also a solution of the boundary value problem .

The introduction of phases which have a priori a fixed order represents a regularization of this shift problem. As start and end point of our cyclic trajectories are always associated with phase boundaries described by equality constraints, those points are uniquely determined on the trajectory, and the shift problem is automatically eliminated.

3.4.5 Solution of Underlying NLP

The discretized problem (3.22) - (3.25) is a nonlinear programming problem (NLP) of large dimension.

Objective functions of Mayer and Lagrange type (3.15) are separable, i.e. they can be written as a sum of functions each valid on a single multiple shooting interval only

$$\tilde{\Phi}(s,q,p) = \sum_{i=0}^{m} \tilde{\Phi}(s_i, q_i, p). \tag{3.28}$$

The discretized variables q_i and s_i have thus only locally restricted influence on the objective function. The same is true for the constraints except for a linear coupling in the continuity and periodicity conditions.

In MUSCOD, the NLP is efficiently solved by a tailored sequential quadratic programming (SQP) method that exploits the problem structure resulting from this local sphere of influence of variables. SQP is an iterative method in which the NLP is locally approximated by quadratic optimization problems (QP). The solution z_k of the QP at the SQP iterate y_k presents the new direction of search for the optimum in the SQP:

$$y_{k+1} = y_k + \alpha z_k \tag{3.29}$$

The QP to be solved is

$$\min_{z_k \in \Omega} \quad \nabla\tilde{\Phi}(y_k)^T z_k + \frac{1}{2} z_k^T B_k(y_k) z_k \tag{3.30}$$

$$\text{s. t.} \qquad h(y_k) + \nabla h(y_k)^T z_k = 0 \tag{3.31}$$

$$\tilde{r}_{eq}(y_k) + \nabla\tilde{r}_{eq}(y_k)^T z_k = 0 \tag{3.32}$$

$$\tilde{r}_{ineq}(y_k) + \nabla\tilde{r}_{ineq}(y_k)^T z_k \geq 0 \tag{3.33}$$

where B_k is an approximation of the Hessian of the Lagrangian function. In a trust region approach Ω is chosen such that the quadratic approximation is valid in the region. The SQP step length α is determined by a line search. For a basic description of SQP methods, see the optimization textbooks Gill et al. [32] and Fletcher [28]. Special partially reduced SQP methods have been developed by Leineweber [48]. In the direct multiple shooting context described in this section the QP has a sparse structure. Instead of directly solving this QP it is favorable to first condense the problem eliminating the additional variables introduced by the multiple shooting method, $s_1, ...s_m$ and then solve the resulting dense QP by a standard solver.

3.4.6 Treatment of Mechanical DAEs

As discussed in section 2.1.1, it is often favorable to formulate mechanical problems as index 3 DAEs instead of ODEs. For our gait models we use this formulation at least for certain phases of the motion. We describe them in the equivalent differentiated index 1 form (2.7) - (2.12) with position and velocity invariants. Those invariants have be taken into account in two different ways:

1. As constraints for the optimization problem: Initial values of integration must lie on the manifold described by the constraints. This is ensured by adding the respective constraints to the set of multipoint constraints at the initial multiple shooting point of the respective phase.

2. As constraints for the integration: To avoid a drift away from this manifold the same constraints would have to be projected onto when integrating the index 1 system instead of the index 3 system. Due to the very short integration intervals in the multiple shooting context drift has so far never appeared to be a problem for our models, and therefore we have not deemed it necessary to use any projection.

In a parallel research project in the group of Bock & Schlöder, Stossmeister [87] pursues a combination of MUSCOD with the integrator libraries MBSSIM and MBSOPT (v. Schwerin & Winckler [95], v. Schwerin [93]) capable to handle index 3 systems by reduction to index 1 and projection onto invariants. As soon as this tool is available, possible effects of this alternative treatment of the drift problem on our models and results can be tested.

Chapter 4

Characterizing the Stability of Periodic Gaits

In chapter 3 we have described numerical methods for the generation of optimal periodic gaits to be used in the inner loop of our two-level optimization procedure. Having found a periodic solution, its stability must be determined as part of the computations in the outer loop. The present chapter is dedicated to appropriate criteria characterizing the stability of a periodic orbit.

Stability of a solution describes the fact that neighboring solutions approach or at least stay close to that solution. Deviations of the physical system's motion from the original precalculated trajectory may occur for a number of reasons. Model parameters can usually only be determined with a certain tolerance. The system's motion is often subject to perturbations caused by the external world. And even the most detailed mathematical model is always some abstraction of the real system. So for the open-loop controlled physical system to be able to automatically recover from this deviated state, the mathematical solution has to be robust against perturbations of the initial values and the parameters.

We start the chapter with mathematical definitions of stability that are needed to explain the theoretical background. We recall the stability theory for linear systems with constant coefficients in section 4.2 and with periodic coefficients – the so-called Floquet theory – in section 4.3. They form the basis for further investigations of nonlinear systems. The theory that we need for our systems is the stability theory for nonlinear periodic systems, which is a special case of Lyapunov's first method. It is introduced in section 4.4. In 4.5, we generalize Lyapunov's first method to our case of periodic systems with discontinuities. Although the method has been applied to gaits before by some authors (e.g. McGeer [52], Coleman [18], Cheng & Lin [15], Hurmuzlu [42]) – varying in the way the relevant quantities are computed – to our knowledge this formal generalization has not been made before. We finish this chapter by briefly recalling the famous second method of Lyapunov and explaining why it is not useful in our investigations. Good references for the first four sections of this chapter are Cronin [22], Walter [98], Meirovitch [56], and Hsu & Meyer [40]. For section 4.6, see La Salle & Lefschetz [80] and Cronin [22].

4.1 Mathematical Definitions of Stability

In this section we give some essential definitions of stability that will be used later in this chapter. The definitions are based on the work of the Russian mathematician A. M. Lyapunov.

A solution $x_0(t)$ of an n-dimensional system of non-autonomous differential equations

$$\dot{x}(t) = f(t, x(t)) \tag{4.1}$$

- is stable (in the sense of Lyapunov), if for each $\epsilon > 0$ there is a $\delta > 0$ such that all solutions $x_1(t)$ with $|x_1(t_0) - x_0(t_0)| < \delta$ satisfy $|x_1(t) - x_0(t)| < \epsilon$ for all $t > t_0$,
- is asymptotically stable (in the sense of Lyapunov), if it is stable and additionally

$$\lim_{t \to \infty} |x_1(t) - x_0(t)| = 0,$$

- is unstable if it is not stable.

Let us recall that for mechanical systems the vector x consists of all position and all velocity variables.

A very useful notion for autonomous systems

$$\dot{x}(t) = f(x(t)) \tag{4.2}$$

especially with closed trajectories is that of orbital stability which only considers the trajectories as entities and not a specific reference point traveling in time along the trajectory.

Let X_0 be the orbit of solution $x_0(t)$ and define $d(X_0, y)$ as the minimum Euclidean distance of a point y from the orbit X_0. Then the solution $x_0(t)$ of an autonomous system

- is orbitally stable if for each $\epsilon > 0$ there is a $\delta > 0$ such that all solutions $x_1(t)$ with $d(X_0, x_1(t_0)) < \delta$ satisfy $d(X_0, x_1(t)) < \epsilon$ for all $t > t_0$, and
- is orbitally asymptotically stable if it is orbitally stable and additionally

$$\lim_{t \to \infty} d(X_0, x_1(t)) = 0.$$

Orbital stability is a considerably weaker condition than stability as points of corresponding values of time t can be far apart, i .e. perturbations along the orbit do not have to be eliminated.

4.2 Stability of Solutions of Linear Systems with Constant Coefficients

As is well known to most engineers and mathematicians, stability of a linear system with constant coefficients can be determined by the eigenvalues of its coefficient matrix. The following theorem can be found e.g. in Walter [98].

Theorem 4.1 *Solution $x \equiv 0$ of*

$$\dot{x}(t) = Ax(t) \quad \textit{with } A \textit{ const.}$$

- *is stable, if $Re(\lambda_i) \leq 0$) for all eigenvalues λ_i of A with $Re(\lambda_i) = 0$ belonging to a non-defective eigenvalue*
- *is asymptotically stable, if $Re(\lambda_i) < 0$) for all eigenvalues λ_i of A*
- *is unstable, if it is not stable.*

Remark:
For general linear systems

$$\dot{x}(t) = A(t)x(t) + f(t) \tag{4.3}$$

where $A(t)$ and $f(t)$ are continuous, it holds that if there exists one solution $x(t; t_0, x_0)$ of (4.3) which is (asymptotically) stable then every solution of (4.3) is (asymptotically) stable. Stability properties of the trivial solution of the linear system can therefore be generalized to arbitrary solutions of the same system.

4.3 Stability of Solutions of Linear Systems with Periodic Coefficients – The Floquet Theory

For time-varying coefficient matrices $A(t)$ it is obviously not possible to derive overall stability properties from the eigenvalues of A as they also change in time.

The theory founded by the French mathematician Gaston Floquet gives an approach for homogenous linear systems with T-periodic coefficients

$$\dot{x}(t) = A(t)x(t) \qquad \text{with} \qquad A(t+T) = A(t). \tag{4.4}$$

The stability of the trivial solution $x \equiv 0$ is studied.

It can be shown that if $x(t)$ is a solution, it follows that $z(t) = x(t+T)$ also is a solution – which does not imply an equivalence of $x(t)$ and $z(t)$. In other words: if $X(t)$ is a fundamental matrix of (4.4) with $X(0) = I$, then $Z(t) = X(t+T)$ is another fundamental matrix and there exists a constant nonsingular matrix C such that

$$X(t+T) = X(t)C \qquad \text{with} \qquad C = X(T). \tag{4.5}$$

The matrix C which is sometimes referred to as monodromy matrix plays an important role in Floquet theory. It is equivalent with the Jacobian of the Poincaré map introduced in section 3.1. The eigenvalues λ_i of C are called characteristic multipliers or Floquet multipliers of A. The monodromy matrix C is not unique but depends on the particular choice of the fundamental matrix $X(t)$. However, the eigenvalues associated with the monodromy matrix are uniquely determined by the system/coefficient matrix $A(t)$ as the monodromy matrices corresponding to different fundamental matrices are similar.

Every fundamental matrix $X(t)$ of (4.4) has a Floquet representation

$$X(t) = Q(t)e^{Bt}, \tag{4.6}$$

where Q(t) is T-periodic and B is constant and satisfies

$$C = X(T) = e^{BT}. \tag{4.7}$$

For every eigenvalue λ_i of C there is a corresponding eigenvalue μ_i of B with identical algebraic multiplicity and

$$\lambda_i = e^{T\mu_i} \tag{4.8}$$

μ_i are called characteristic exponents or Floquet exponents of A. From the definition of B follows that the real parts of μ_i are uniquely determined, the imaginary parts are defined up to an integral multiple of $2\pi/T$.

Transformation (4.6) allows to derive stability statements for systems with periodic coefficients from systems with constant coefficients.

Theorem 4.2 *Solution* $x \equiv 0$ *of*

$$\dot{x} = A(t)x \quad \textit{with} \quad A(t+T) = A(t)$$

- *is stable, if all characteristic multipliers* $|\lambda_i| \leq 1$ *(or all characteristic exponents* $Re(\mu_i) \leq 0$*) with* $|\lambda_i| = 1$ *(*$Re(\mu_i) = 0$*) belonging to a non-defective eigenvalue,*
- *is asymptotically stable, if all multipliers* $|\lambda_i| < 1$ *(or all exponents* $Re(\mu_i) < 0$*), and*
- *is unstable, if at least one multiplier* $|\lambda_i| > 1$ *(or* $Re(\mu_i) > 0$*).*

The theorem can be found in Walter [98].

T-periodic solutions of those systems with T-periodic coefficients exist, if the transfer matrix C has one eigenvalue $\lambda = 1$. If no characteristic multiplier equals one, then equation (4.4) has no nontrivial solution of period T.

4.4 Stability of Periodic Solutions of Nonlinear Systems – Lyapunov's First Method

Lyapunov's first method is a stability theory for nonlinear systems that is based on approximations by the corresponding linear systems.

Here we only treat T-periodic nonlinear systems

$$\dot{x} = f(t, x) \quad \text{with} \quad f(t + T, \cdot) = f(t, \cdot) \tag{4.9}$$

where $f \in \mathcal{C}^2$. We also consider the special case of an autonomous equation

$$\dot{x} = f(x) \tag{4.10}$$

which is trivially T-periodic.

Periodic solutions of (4.9) and (4.10) do not necessarily have to exist. Assuming that there exists a periodic solution $x_p(t + T) = x_p(t) = x_p(t; t_0, x_{0,p})$, then any other solution x_1 can be expressed as

$$x_1(t) = x_p(t) + \Delta x \tag{4.11}$$

Since x_1 is a solution of (4.9)

$$\dot{x_p} + \dot{\Delta x} = f(t, x_p + \Delta x) \tag{4.12}$$

holds and it follows that

$$\dot{\Delta x} = f(t, x_p + \Delta x) - f(t, x_p). \tag{4.13}$$

By means of a Taylor series expansion this can be written as

$$\dot{\Delta x} = \frac{\partial f}{\partial x}(t, x_p)\Delta x + h(t, \Delta x) \tag{4.14}$$

with

$$\frac{\partial f}{\partial x}(t, x_p) = \left[\frac{\partial f_i}{\partial x_j}(t, x_p)\right] \tag{4.15}$$

Equation (4.14) is the variational system of (4.9) relative to solution $x(t)$. The linear equation

$$\dot{\Delta x} = \frac{\partial f}{\partial x}(t, x_p)\Delta x \tag{4.16}$$

is called the linear variational system (Cronin [22]). Note that T-periodicity of f causes T-periodicity of the derivative matrix $\left[\frac{\partial f_i}{\partial x_j}\right]$ but not necessarily of the solution Δx.

The theory of stability for solutions of nonlinear differential equations is based on the study of the trivial solution of this linear variational system. For this we need the results of Floquet's theory described in the previous section.

If $\Delta x \equiv 0$ is a stable solution of the linear variational system then solution x_p of (4.9) is called infinitesimally stable. The question is now if infinitesimal stability implies stability. For periodic and constant matrices $\left[\frac{\partial f_i}{\partial x_j}\right]$ conjectures about asymptotic stability and instability can be made. For this we need the monodromy matrix C which in the case of nonlinear systems is defined by

$$C(t, t + T) = \left[\frac{\partial x(t + T)}{\partial x(t)}\right]. \tag{4.17}$$

For the following two theorems, see Hsu & Meyer [40].

Theorem 4.3 (Stability of periodic solutions of non-autonomous systems) *For a non-autonomous system $\dot{x}(t) = f(t, x(t))$ with $f(t, \cdot) = f(t+T, \cdot)$ the variational equation about a periodic solution $x_p(t) = x_p(t+T), T \neq 0$ is given by*

$$\dot{\Delta x} = \frac{\partial f}{\partial x}(t, x_p)\Delta x + h(t, \Delta x).$$

It is assumed that

$$\lim_{||\Delta x|| \to 0} \frac{||h(t, \Delta x)||}{||\Delta x||} = 0.$$

Then the periodic solution $x_p(t)$ is asymptotically stable if $|\lambda_i| < 1$ for all eigenvalues λ_i of the monodromy matrix $C(t, t+T)$.

If at least one eigenvalue $|\lambda_i| > 1$, the system is unstable. If one eigenvalue is exactly one, the system is said to exhibit critical behavior. No conclusions about stability can be drawn from the linear study: depending on higher order terms the system can be either stable or unstable.

Note that periodic solutions of nonlinear systems are not generally associated with eigenvalues of one as in the linear case because the solution of the linear variational system does not necessarily have to be periodic as noted above.

This is different in the autonomous case as can be shown by substituting a periodic solution into (4.10) and differentiating with respect to time

$$\frac{d}{dt}\left(\frac{dx_p}{dt}\right) = \frac{d}{dt}f(x_p) = \frac{df_p}{dx}(x_p)\frac{dx_p}{dt} \tag{4.18}$$

Comparison with (4.16) shows that $\frac{dx_p}{dt}$ is a solution of the linear variational system. As this solution is periodic, the monodromy matrix $C(t, t+T)$ of the autonomous system has at least one eigenvalue of one. So theorem 4.3 can not be applied to autonomous systems. All we can ask for in this case is orbital stability.

Theorem 4.4 (Stability of periodic solutions of autonomous systems) *For an autonomous system $\dot{x} = f(x)$ the variational equation about a periodic solution $x_p(t) = x_p(t+T), T \neq 0$ is given by*

$$\dot{\Delta x} = \frac{\partial f}{\partial x}(x_p)\Delta x + h(t, \Delta x).$$

It is assumed that

$$\lim_{||\Delta x|| \to 0} \frac{||h(\Delta x)||}{||\Delta x||} = 0.$$

Then the periodic solution $x_p(t)$ is stable and orbitally asymptotically stable if $|\lambda_i| < 1$ for all eigenvalues λ_i of the monodromy matrix $C(t, t+T)$ except for one eigenvalue $|\lambda_k| = 1$.

Note that in both cases, for autonomous as well as for non-autonomous systems, stability in terms of the eigenvalues of the monodromy matrix is not affected by the choice of the starting point of the periodic sample. Monodromy matrices for the intervals $[t_0, t_0 + T]$ and $[t_1, t_1 + T]$ are different for $t_0 \neq t_1$ but their eigenvalues are the same, since they are related by a similarity transformation

$$C(t_0, t_0 + T) = WC(t_1, t_1 + T)W^{-1} \tag{4.19}$$

4.5 Generalization of Lyapunov's First Method to Discontinuous Periodic Systems

For $\mathcal{C}^2$-continuous problems, Lyapunov's first method allows one to draw conclusions about asymptotic stability, or instability, respectively, of the solution of the nonlinear equations. However, our systems are nonlinear multi-phase problems with discontinuities, and do not satisfy the requirements of the standard results in stability theory.

The purpose of this section is to show that one can nevertheless conclude asymptotic stability of the gait solution of the nonlinear equations from the asymptotic stability of the linear map. More specifically, we want to prove the following theorem:

Theorem 4.5 *We study a non-autonomous T-periodic system with multiple phases, that is piecewise $\mathcal{C}^2$-continuous but has discontinuities J_j between phases*

$$\begin{aligned} \dot{x}(t) &= f_j(t, x(t)) \quad \text{for } [t_{j-1}, t_j] \text{ with } j = 1, ..., n_{ph} \text{ and } t_0 = 0, t_{n_{ph}} = T \\ &\qquad \text{and} \quad f_j(t, \cdot) = f_j(t + T, \cdot),\ f_j \in \mathcal{C}^2 \qquad (4.20) \\ s_j(t_j, x(t_j)) &= 0 \qquad (4.21) \\ x(t_j^+) &= x(t_j^-) + J_j(t_j, x(t_j^-)) \qquad (4.22) \end{aligned}$$

and a T-periodic solution $x_p(t) = x_p(t + T)$. It is assumed that

- *the solution x_p at any instant $\tilde{t}$ is twice continuously differentiable with respect to initial values*
- *the divergence from the base solution caused by a perturbation $||\Delta x||$ is correctly described by a linearization up to first order, i.e. with an error that is quadratic in $||\Delta x||$.*

Then a monodromy matrix $C(t, t + T)$ can be defined and the periodic solution x_p is asymptotically stable if $|\lambda_i| < 1$ for all eigenvalues of $C(t, t + T)$.

Proof:
We start by showing that the assumptions of the theorem are valid in the case of the gait models studied in this thesis. Overall discontinuous but piecewise continuous functions can under certain conditions exhibit sufficient differentiability properties with respect to initial values. In analogy to theorem 3.1 of Bock[8] that has been formulated for boundary value problems it can be shown for initial value problems with discontinuous right hand

side, that a solution F is k times continously differentiable with respect to initial values if

- the right hand side f has a finite number of switching points t_s which are isolated, i.e. roots of different switching functions do not coincide, and well defined, i.e. the derivative $\dot{s}(t_s^-)$ exists at all those points and satisfies

$$\dot{s}(t_s^-) > 0 \qquad \text{if } s(t_s - \epsilon) < 0 \tag{4.23}$$
$$\dot{s}(t_s^-) < 0 \qquad \text{if } s(t_s - \epsilon) > 0. \tag{4.24}$$

  In other words, the roots t_s of the switching points have to be simple roots and do not degenerate in the presence of perturbations.
- F is k times piecewise continuously differentiable w.r.t. the initial values of the respective piece, i.e. the right hand sides f_j are $k-1$ time piecewise continuously differentiable w. r. t. initial values
- switching functions and right hand sides must be extendable beyond switching points
- all switching functions and jump functions are k times piecewise continuously differentiable.

The theorem requires the solutions to be twice continuously differentiable with respect to initial values. All our gait models satisfy the above stated conditions for at least $k = 2$.

Denoting the solution of the initial value problem (4.20) with $x(0) = y$ at $t = T$ as $F(y, T) =: F(y)$, the second assumption is stated as

$$||F(x + \Delta x) - F(x) - F_x(x)\Delta x|| \leq b||\Delta x||^2. \tag{4.25}$$

F_x is the derivative of F with respect to initial values, and b is a constant. The left hand side of relation (4.25) can be transformed, using the Jacobian F_x

$$\begin{aligned} & ||F(x + \Delta x) - F(x) - F_x(x)\Delta x|| \\ = \; & || \int_0^1 F_x(x + \alpha\Delta x)\Delta x \; d\alpha - \int_0^1 F_x(x)\Delta x \; d\alpha \; || \\ = \; & || \int_0^1 \left(F_x(x + \alpha\Delta x) - F_x(x)\right) \Delta x \; d\alpha \; || \\ = \; & \int_0^1 || \left(F_x(x + \alpha\Delta x) - F_x(x)\right) \Delta x|| \; d\alpha. \end{aligned} \tag{4.26}$$

According to the mean value theorem, we have for each component of a continuously differentiable Jacobian F_x – twice continuously differentiable F –

$$\frac{F_{x,ij}(x + \alpha\Delta x) - F_{x,ij}(x)}{||\alpha \cdot \Delta x||} = \frac{dF_{x,ij}}{d\alpha}(x + \xi_{ij}\alpha\Delta x) \text{ for some } \xi \in [0, 1]. \tag{4.27}$$

The right hand side is bounded by some constant $\omega_{i,j}$ with a maximum value over all components of the Jacobian

$$\omega_{i,j} \leq \omega. \tag{4.28}$$

Therefore,

$$\frac{||\left(F_x(x+\alpha\Delta x)-F_x(x)\right)||}{\alpha||\Delta x||} \leq \tilde{\omega} < \infty \tag{4.29}$$

or

$$||\left(F_x(x+\alpha\Delta x)-F_x(x)\right)|| \; ||\Delta x|| \leq \tilde{\omega}\; \alpha||\Delta x||^2. \tag{4.30}$$

With this, we can estimate

$$\begin{aligned} (4.26) \quad &\leq \int_0^1 ||\left(F_x(x+\alpha\Delta x)-F_x(x)\right)|| \; ||\Delta x|| d\alpha \\ &\leq \int_0^1 \tilde{\omega}\alpha||\Delta x||^2 d\alpha \\ &= \frac{1}{2}\tilde{\omega}||\Delta x||^2. \end{aligned} \tag{4.31}$$

Choosing the constant $b = \frac{\tilde{\omega}}{2}$, this completes the proof of statement (4.25).

The monodromy matrix $C(t, t+T)$ is equal to the Jacobian $F_x(x_p) = F(x_p, T)$ of a periodic solution x_p). It is computed by a matrix multiplication from the transfer matrices $C(t_{j-1}, t_j)$ of the individual phases j and 'update matrices' describing the derivatives

$$\frac{\partial x_j^+}{\partial x_j^-} = I + \frac{\partial J_j}{\partial x_j^-} =: U(t_j) \tag{4.32}$$

at a discontinuous point. See section 6.4 for details about the computation of monodromy matrices. Formulas for the computation of derivatives of the monodromy matrix, the existence of which has been shown above, are given in section 6.6. Matrices $U(t_j)$ are regular if phases $j-1$ and j of the gait have the same degrees of freedom, and singular otherwise. Monodromy matrices $C(t_{j-1}, t_j)$ of all phases are regular.

From (4.25) we can conclude that

$$\begin{aligned} ||F(x_p+\Delta x)-F(x_p)|| &\leq ||C(t,t+T)\Delta x|| + b||\Delta x||^2 && (4.33) \\ &\leq ||C(t,t+T)||\; ||\Delta x|| + b||\Delta x||^2. && (4.34) \end{aligned}$$

If $|\lambda_i| < 1$ for all eigenvalues of C, then the norm $||\cdot||$ can be chosen such that $C(t, t+T) \leq \kappa < 1$ (compare theorem 5.2). Perturbations therefore decay in this norm, and the periodic solution is asymptotically stable.

It remains to be shown for the case of piecewise continuous but overall discontinuous solutions that the eigenvalues of the monodromy matrix $C(t, t+T)$ do not depend on the starting time t of the period. This is equivalent with showing that for general square matrices $A, B \in \mathbb{R}^{n\times n}$

$$\lambda_i(AB) = \lambda_i(BA) \quad i = 1, ..., n. \tag{4.35}$$

For regular matrices this follows from similarity transformations. The proof for general square matrices is given in Wilkinson [101]. It even can be shown for nonsquare matrices $A \in \mathbb{R}^{n\times m}$ and $B \in \mathbb{R}^{m\times n}$ that AB and BA have the same eigenvalues except that the

product which is of higher order has $|n-m|$ additional zero eigenvalues. From this we can conclude that the eigenvalues remain unchanged even if the phases are described by different numbers of equations. □

4.6 Some Words about Lyapunov's Second Method

More famous than the above described first method of Lyapunov is his second or direct method which is of great use in analytical dynamics. We briefly introduce this method as it simply can not be omitted in a chapter discussing the stability of motion and then explain why we can not use this method for our intentions.

It is assumed that $x \equiv 0$ is a solution of the nonlinear differential equation (4.9), i. e.

$$0 = f(t, 0) \tag{4.36}$$

In contrast to the method described in the previous sections, Lyapunov's second method does not rely on a study of the linear parts of the equation.

Instead, the method demands the construction of a so called Lyapunov function.

The idea behind that is that, according to Lagrange's theorem, the potential energy of a physical system is minimal at a stable equilibrium point and maximal at an unstable equilibrium. The Lyapunov function $V(t, x)$ represents a generalization of the potential energy function.

It has the domain $D_v = \{(t, x) | t > t_1, |x| < A\}$ and must exhibit the following properties:

- continuous first partial derivatives with respect to t and x_i : $V(t, x) \in \mathcal{C}^1(D)$
- $V(t, 0) = 0$ for $t > t_1$
- positive definiteness: $V(t, x) > 0$ for $x \neq 0$
- negative definiteness of derivative: $\dot{V}(t, x) \leq 0$.

The derivative $\dot{V}(t, x)$ which is the derivative of $V(t, x)$ along the solution $x(t)$ is defined as

$$\dot{V}(t, x(t)) = \sum_{i=1}^{n} \frac{\partial V}{\partial x_i} \dot{x}_i + \frac{\partial V}{\partial t} \tag{4.37}$$

According to the second method, the existence of such a Lyapunov function proves the stability of the trivial solution of the system. In detail, we can distinguish:

- $\dot{V}(x) \leq 0$ in D $\rightarrow$ stability (in the sense of Lyapunov),

- $\dot{V}(x) < 0$ in D $\rightarrow$ asymptotic stability,
- $\dot{V}(x) \leq -\alpha V$ and $V(x) \geq b|x|^{\beta}$ in D with $(\alpha, \beta, b > 0)$ $\rightarrow$ exponential stability.

The difficulty of the application of this method to a physical system consists in the construction of an appropriate Lyapunov function. Such functions have only been found for certain classes of systems, e.g. the total energy is a Lyapunov function for Hamiltonian systems. There are no general rules for the construction of Lyapunov functions such that the process can not be automated. It is therefore not suited for the use in an optimization environment which has to be applied to very different systems of difficult discontinuous type.

Chapter 5

Stability as Non-standard Optimization Criterion

The stability criterion based on the maximum eigenvalue of a monodromy matrix has been discussed extensively in the previous chapter. In this chapter now, it is outlined that using this criterion as objective function clearly leads to a non-standard optimization problem.

Section 5.1 describes the various difficulties of stability optimization in terms of eigenvalue optimization of the monodromy matrix formulated as two-level or one-level optimization problem. In section 5.2 we discuss possible alternatives for the maximum eigenvalue function to be used in stability optimization. The study of powers of the monodromy matrix instead of the matrix itself is discussed in section 5.3. In section 5.4 we finally summarize the stability optimization criteria that we are using in our computations.

Basic knowledge about eigenvalue problems is assumed in this chapter. Good references for eigenvalue theory are the classical textbook of Wilkinson [101], and, for the more numerical point of view, the books of Trefethen & Bau [92] and of Demmel [24]. Since the monodromy matrix is a real non-symmetric matrix, our considerations focus on that case, sometimes pointing out differences to the simpler case of a symmetric matrix.

5.1 Difficulties of Stability Optimization in Terms of Eigenvalue Optimization

As we have seen in the previous chapter, all real and complex eigenvalues of the monodromy matrix have to lie inside the unit circle for the system to be stable. This goal can be achieved by minimizing the largest eigenvalue by magnitude $|\lambda_{max}|$ (which is also called the spectral radius ρ) of the monodromy matrix C

$$\min_x |\lambda_{max}(C(x))| = \min_x |\rho(C(x))| \tag{5.1}$$

hoping that the minimum will have a spectral radius smaller than one. x is the vector of all free optimization variables to be specified later. This objective function is a concatenation of two functions $\rho(C) \circ C(x)$: first, the matrix C is determined as a function of x, secondly, the spectral ρ radius of the matrix C is determined. The first part of this objective function smooth but generally non-convex, whereas the second part is convex but generally non-smooth.

Stability optimization in terms of minimizing the spectral radius is a difficult optimization problem for several reasons. To structure our considerations, we split those difficulties into three different groups to be treated independently in the following subsections:

- difficulties due to the minimization of the spectral radius of arbitrary matrices, i.e. the dependency of ρ on C,
- difficulties caused by the nature of the matrix itself, i.e. the dependency of C on x,
- difficulties caused by aspects of the optimization problem other than this objective function like constraints imposed by the dynamics.

5.1.1 Minimizing the Maximum Eigenvalue of a Non-symmetric Matrix

Eigenvalues of a matrix are the roots of its characteristic polynomial. For matrix dimensions n larger than four no analytic solution is possible, so every eigenvalue solver must be iterative. Typically, eigenvalue solvers for non-symmetric matrix need $O(n^3)$ operations (see Trefethen & Bau [92]). This indicates that functions involving eigenvalues are not computationally cheap, but as we will see later, the really expensive part in our case lies in the computation of the matrix C itself.

Eigenvalues are continuous functions of the matrix entries but they are non-differentiable at points where they coalesce. Two or more equal eigenvalues may not be the natural case for a physical system, but minimizing the maximum eigenvalue tends to make all eigenvalues equal (at least as far as it is allowed by the constraints).

Before we look at those points of multiple eigenvalue in more detail we need to introduce some important facts about the relationship between non-symmetric matrices and their eigenvalues.

Assuming that we have a simple eigenvalue λ of a non-symmetric matrix C, a perturbation of C by δC leads to a perturbation of the eigenvalue by $\delta\lambda$:

$$\delta\lambda = \frac{v_l^T \delta C v_r}{v_l^T v_r} + O(||\delta C||^2) \tag{5.2}$$

or, for the absolute value

$$|\delta\lambda| \leq \frac{||\delta C||}{|v_l^T v_r|} + O(||\delta C||^2) \tag{5.3}$$

where v_r and v_l are the right and left normalized eigenvectors associated with the eigenvalue λ.

Proof: Subtracting

$$Cv_r = \lambda v_r$$

from the equation for the perturbed eigenvalue

$$(C + \delta C)(v_r + \delta v_r) = (\lambda + \delta\lambda)(v_r + \delta v_r),$$

ignoring second order terms and multiplying the result by v_l^T from the left, leads to:

$$v_l^T(C\,\delta v_r + \delta C\,v_r) = v_l^T(\delta\lambda\,v_r + \lambda\,\delta v_r).$$

As $v_l^T C = v_l^T \lambda$, this is reduced to

$$v_l^T\,\delta C\,v_r = v_l^T\,\delta\lambda\,v_r$$

and hence

$$\delta\lambda = \frac{v_l^T\,\delta C\,v_r}{v_l^T v_r}.$$

□

The condition number *cond* of the eigenvalue is

$$cond(\lambda) = |v_l^T v_r|^{-1}, \tag{5.4}$$

which is the secans of the angle between left and right eigenvector.

The derivative of a simple eigenvalue with respect to the matrix entries is computed in analogy to equation (5.2)

$$\frac{d\lambda}{dC} = \left[\frac{d\lambda}{dc_{ij}}\right] = \frac{v_l v_r^T}{v_l^T v_r} \tag{5.5}$$

Remark:
Note that for a symmetric matrix where left and right eigenvectors are equal $v := v_l = v_r$ with $|v| = 1$, things would be much simpler. An eigenvalue perturbation is computed by

$$\delta\lambda = \frac{v^T\delta C v}{v^T v} + O(||\delta C||^2) = v^T\delta C v + O(||\delta C||^2) \tag{5.6}$$

and

$$|\delta\lambda| \leq ||\delta C|| + O(||\delta C||^2), \tag{5.7}$$

and the condition number of eigenvalues of symmetric matrices is always one. The derivative of simple eigenvalues with respect to matrix entries in this case becomes

$$\frac{d\lambda}{dC} = vv^T. \tag{5.8}$$

Now we can resume the study of the points with multiple maximum eigenvalue (by magnitude) with multiplicity m for which three different types exist. We list the properties of all types and their consequences for possible numerical algorithms.

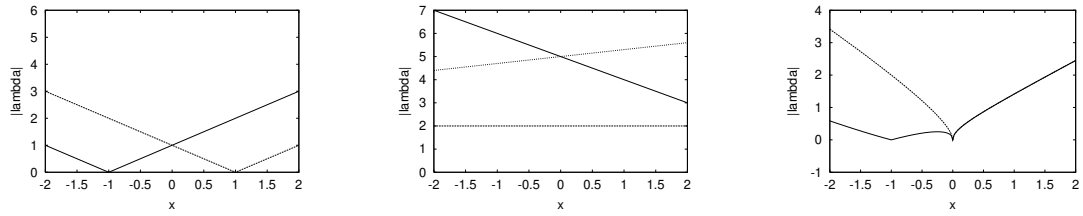


Figure 5.1: Three different types of multiple maximum eigenvalue modulus at x=0 for one-dimensional dependency: a) distinct eigenvalues, but same magnitude, b) non-defective multiple eigenvalue c) defective multiple eigenvalue

a) Eigenvalues are distinct, but have the same magnitude:

In this case all individual eigenvalues are differentiable, and the derivatives can be computed according to formula (5.5). This is also true for the two eigenvalues of a conjugate complex couple. The spectral radius is simply the pointwise maximum of several differentiable functions. Figure 5.1a shows an example of this type:

$$A_a = \begin{pmatrix} 1+x & 1 \\ 0 & -1+x \end{pmatrix}$$

Treatment with standard min-max techniques is possible (see e.g. Gill et al. [32]). The original problem

$$\min_x \left(\max_{f_i} (f_1(x), ..., f_m(x)) \right) \tag{5.9}$$

is reformulated by introducing a new variable y

$$\min_{x,y} \quad y \tag{5.10}$$

$$s.t. \quad f_i(x) \leq y \quad i = 1, ..., n. \tag{5.11}$$

If several f_i attain the maximum then simply more than one of the inequality constraints become active. This resulting problem can be solved by standard methods for constrained nonlinear optimization problems.

b) Maximum eigenvalue is multiple, but non-defective:

Geometric and algebraic multiplicity of the eigenvalue are the same, and the corresponding eigenvector space has full dimension. However, the individual eigenvectors that span the space are not uniquely defined. The matrix is still diagonalizable (assuming, of course, that there is no other set of multiple eigenvalues causing trouble). As an example for this type, matrix

$$A_b = \begin{pmatrix} 5-x & 0 & 0 \\ 0 & 5+0.3x & 1 \\ 0 & 0 & 2 \end{pmatrix}$$

is shown in figure 5.1b. The individual eigenvalues are non-differentiable at the points of multiple eigenvalue due to the non-uniqueness of eigenvectors, but they have a finite condition number and are Lipschitz continuous.

As the eigenvalues are always well-conditioned and differentiable in the neighborhood of the points of multiple eigenvalue, and as this point only forms a singularity in an otherwise continuously differentiable manifold, the derivative at this point can be 'reconstructed' by applying a small perturbation to the matrix and computing the derivative at this perturbed point.

c) Maximum eigenvalue is multiple and defective:

Algebraic multiplicity m of maximum eigenvalue exceeds its geometric multiplicity m_g, such that the eigenvector space does not have full dimension. The matrix is non-diagonalizable, and the respective Jordan form would have m_g blocks containing the defective multiple eigenvalues. An example for this type,

$$A_c = \begin{pmatrix} x & 1 \\ -x & x \end{pmatrix}$$

is shown in figure 5.1c. This type only exists for non-symmetric matrices, as symmetric ones are always diagonalizable.

The right and left eigenvectors of a Jordan block of dimension m are e_1 and e_m and thus perpendicular, such that the condition number is always infinite at those points (compare equation (5.4)). Infinite condition number does not mean that multiple eigenvalues cannot be computed with any accuracy at all. Instead, one can expect to correctly compute $1/m$ of the machine precision digits for an eigenvalue with multiplicity m and a matrix given with machine precision (Demmel [24]). But infinite condition number or perpendicular left and right eigenvectors also cause the derivative of the eigenvalues to be infinite (equation (5.5), i.e. the eigenvalues are not even Lipschitz at the points of multiple eigenvalue. Typically some bifurcation occurs at this point; for the example shown in figure 5.1 two distinct real eigenvalues for $x < 0$ 'join' to a conjugate complex couple for $x > 0$.

The eigenvalues are also ill-conditioned in the neighborhood as the eigenvectors continuously approach perpendicularity.

Due to the properties stated above any gradient based algorithm would encounter difficulties not only in the singular points themselves but also in their neighborhood. We have to expect that it would not be possible to compute meaningful derivatives in a region about these points.

Figure 5.2 visualizes the spectral radius manifold for a simple 2-dimensional matrix depending on two variables. It shows several local minima all lying at points with multiple eigenvalue, most of them of type c). This gives a little hint about the difficulties that can arise for matrices which are not that simple any more.

In section 6.1.2 we will give an overview of existing algorithms in the literature for simple cases of eigenvalue optimization for analytic matrices.

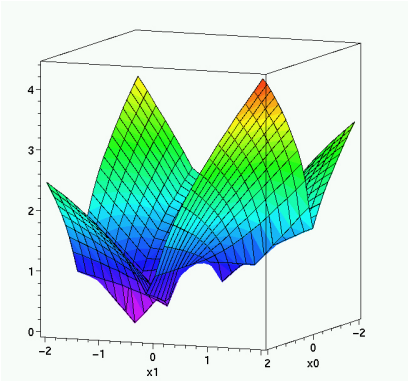


Figure 5.2: Absolute value of maximum eigenvalue of matrix $((x_1, x_0), (2x_1, 1))$

5.1.2 Computation of the Monodromy Matrix

The monodromy matrix of a periodic trajectory is obviously not a simple analytic function of the variables x. It contains the derivatives of the trajectory end values with respect to initial values. This makes the objective function less accurate than an analytic function and computationally very expensive since for every function evaluation a complete periodic trajectory plus the derivatives have to be computed. Additionally, since the objective function already contains first order derivatives of the trajectory in terms of the matrix C, its gradients would even need second derivatives in terms of $\frac{dC}{dx_0}$ and $\frac{dC}{dp}$. Those gradients are even more expensive to compute, and it should be kept in mind that they do not necessarily satisfy the accuracy demands of some algorithms. The monodromy matrix typically is a non-convex 'function' of the independent variables.

5.1.3 Constraints of Stability Optimization

Stability optimization represents not only an eigenvalue optimization problem with a matrix difficult to compute as demonstrated in the two previous sections. Differential equations, periodicity and switching functions etc. also enter as constraints to eigenvalue optimization.

Two-level Formulation

When treated as two-level optimization problem as we do in this thesis, the outer loop stability optimization is formulated as an unconstrained problem. But every function evaluation in the outer loop includes a determination of a periodic gait for the given set of parameters by the methods described in chapter 3. This causes function evaluations of the outer loop to be very expensive and potential gradient computations to be rather difficult.

The full problem formulation in this case contains a non-standard parameter optimization problem in the outer loop and a standard optimal control problem in the inner loop:

$$\min_{p} \quad |\lambda_{max}(C_x)| \tag{5.12}$$

with C_x being the monodromy matrix of the solution of

$$\min_{x,u,T} \quad \int_0^T \phi(x(t),u(t),p)\,dt \quad + \quad \Phi(T,x(T),p) \tag{5.13}$$

$$\begin{aligned}
\text{s. t.} \qquad \dot{x}(t) &= f_j(t,x(t),u(t),p) \quad \text{for} \quad t\in[\tau_{j-1},\tau_j], \\
&\qquad\qquad j=1,...,n_{ph},\ \tau_0=0,\tau_{n_{ph}}=T && (5.14)\\
g_j(t,x(t),u(t),p) &\geq 0 \qquad \text{for} \quad t\in[\tau_{j-1},\tau_j] && (5.15)\\
x(\tau_j^+) &= h(x(\tau_j^-)) \quad \text{for} \quad j=1,...,n_{ph} && (5.16)\\
r_{eq}(x(0),..,x(T),p) &= 0 && (5.17)\\
r_{ineq}(x(0),..,x(T),p) &\geq 0. && (5.18)
\end{aligned}$$

Its solution requires methods for non-standard, non-differentiable parameter optimization.

One-level Formulation

In a one-level formulation the system's dynamics as well as all other constraints imposed for periodic gait generation become constraints of stability optimization to be satisfied simultaneously. This leads to the formulation of a non-standard optimal control problem:

$$\min_{x,u,p,T} \quad |\lambda_{max}(C_x)| \tag{5.19}$$

$$\begin{aligned}
\text{s. t.} \qquad \dot{x}(t) &= f_j(t,x(t),u(t),p) \quad \text{for} \quad t\in[\tau_{j-1},\tau_j], \\
&\qquad\qquad j=1,...,n_{ph},\ \tau_0=0,\tau_{n_{ph}}=T && (5.20)\\
x(\tau_j^+) &= h(x(\tau_j^-)) \quad \text{for} \quad j=1,...,n_{ph} && (5.21)\\
g_j(t,x(t),u(t),p) &\geq 0 \qquad \text{for} \quad t\in[\tau_{j-1},\tau_j] && (5.22)\\
r_{eq}(x(0),..,x(T),p) &= 0 && (5.23)\\
r_{ineq}(x(0),..,x(T),p) &\geq 0. && (5.24)
\end{aligned}$$

In contrast to the problems treated in chapter 3, its objective function is not of Mayer or Lagrange type and cannot be transformed into one of those. There is no way to express the spectral radius of the overall monodromy matrix as a sum of functions depending only on local variables of the respective multiple shooting interval. This problem asks therefore for special solution methods for non-standard optimal control problems that can not rely on the same assumptions of structure as the methods described in chapter 3.

5.2 Alternative Objective Functions for Stability Optimization

The purpose of this section is to study alternatives that could help to make stability optimization a problem easier to solve. The approaches presented here all aim at replacing the minimization of the maximum eigenvalue modulus by some other optimization criterion to avoid the difficulties detailed in section 5.1.1.

5.2.1 Shortcomings of Obvious Ideas

We briefly describe some simple but misleading approaches that have been proposed to avoid the difficulties of eigenvalue optimization and give reasons why they are not applicable.

The first idea is to treat min-max eigenvalue problems by standard min-max problem reformulation into a nonlinear optimization problem. But let us recall from section 5.1.1, that except for points of multiple maximum eigenvalue of type a) the individual eigenvalues become non-differentiable at those points. The typical ill-conditioning in the region of a point of type c) and infinite derivatives also may not be handled by standard nonlinear programming methods. Another problem is that the different eigenvalues of a matrix cannot be labeled in the sense that corresponding eigenvalues at different iterates cannot be uniquely identified. So it would be impossible to always associate one constraint with 'the same' eigenvalue, and a switching of eigenvalues between constraints might take place.

The same problems (local non-differentiability, ill-conditioning and switching of eigenvalues) would be encountered if instead of using an objective function based on the maximum eigenvalue one chose to impose constraints on all eigenvalues

$$|\lambda_i| < 1 - \delta \quad i = 1, ..., n \tag{5.25}$$

with some $\delta > 0$, e.g. $\delta = 0.1$.

Another idea is to use an auxiliary objective function that takes all eigenvalues into account and punishes eigenvalues outside the unit circle, e.g.:

$$\min f = \sum_i f_i(\lambda_i) \quad \text{with} \quad f_i(\lambda_i) = \arctan(10(|\lambda_i| - 1)) \tag{5.26}$$

Again, this does not solve most of the problems listed in section like non-differentiability and ill-conditioning.

5.2.2 Equivalence of Norms & the Theorem of Hirsch

In order to overcome these mathematical difficulties a far better idea is to look for some well-behaving function that does not directly depend on the eigenvalues but is known to

be an upper bound to the spectral radius. Thus one could expect to reduce the maximum eigenvalue by minimizing this function. The so called induced matrix norms (see e.g. Stoer [85] or Trefethen & Bau [92]) have the property of being an upper bound to the spectral radius as detailed below.

Induced matrix norms are always derived from corresponding vector norms. They describe the effect of a matrix $C \in \mathbb{C}^{n_1 \times n_2}$ as an operator between domain space of dimension n_2 and range space of dimension n_1. We will concentrate on real, square matrices $C \in \mathbb{R}^{n \times n}$. The induced matrix norm $||C||$ is defined as the largest factor by which a vector is "multiplied" by the matrix C:

$$||C|| = \max_{x \neq 0} \frac{||Cx||}{||x||}. \tag{5.27}$$

In this sense the maximum norm of a matrix C with entries c_{ij} becomes the maximum row sum:

$$||C||_\infty = \max_{x \neq 0} \frac{||Cx||_\infty}{||x||_\infty} = \max_{x \neq 0} \frac{\max_i |\sum_{j=1}^n c_{ij} x_j|}{\max_j |x_j|} = \max_i \sum_{j=1}^n |c_{ij}|, \tag{5.28}$$

the 1-norm is the maximum column sum:

$$||C||_1 = \max_{x \neq 0} \frac{||Cx||_1}{||x||_1} = \max_{x \neq 0} \frac{\sum_{i=1}^n (\sum_{j=1}^n c_{ij} x_j)}{\sum_{j=1}^n |x_j|} = \max_j \sum_{i=1}^n |c_{ij}|, \tag{5.29}$$

and the Euclidean norm equals to the largest singular value of C

$$\begin{aligned} ||C||_2 &= \max_{x \neq 0} \frac{||Cx||_2}{||x||_2} = \max_{x \neq 0} \frac{\sqrt{x^T C^T C x}}{\sqrt{x^T x}} = \sqrt{\lambda_{max}(C^T C)} \\ &= \sigma_{max}(C). \end{aligned} \tag{5.30}$$

Based on these definitions of induced matrix norms we now can state the following theorems (Stoer & Bulirsch [86]):

Theorem 5.1 (Theorem of Hirsch)
All eigenvalues λ of any given matrix C satisfy:

$$|\lambda| \leq ||C|| \tag{5.31}$$

In other words: any induced norm of the matrix C is an upper bound to the spectral radius

$$|\rho| \leq ||C||. \tag{5.32}$$

Theorem 5.2
1. For each matrix C and each ϵ there is a vector norm $||\cdot||$ and a corresponding matrix norm for which

$$||C|| \leq \rho(C) + \epsilon. \tag{5.33}$$

2. If the maximum eigenvalue λ with $|\lambda| = \rho$ is non-defective, then there even is an induced matrix norm for which

$$||C|| = \rho(C). \tag{5.34}$$

Hence the spectral radius of a matrix being the infimum of all induced matrix norms is equivalent to or at least very close to some induced matrix norm, but in general this specific norm is neither physically nor mathematically relevant.

As it will be discussed in the following sections all three induced matrix norms presented above (1-, 2-, and ∞-norm) are better behaving functions than the spectral radius.

Let us recall an important property of norms in $\mathbb{R}^n$: in finite-dimensional vector space, all norms are equivalent in the sense that if $||\cdot||_{q_1}$ and $||\cdot||_{q_2}$ are two norms on the same space, then there exist positive constants c_1 and c_2 such that for all x in that space

$$c_1||x||_{q_1} \leq ||x||_{q_2} \leq c_2||x||_{q_1}. \tag{5.35}$$

From that and the above theorems it follows that the maximum eigenvalue and some induced matrix norm are connected by some finite factor, but of course the norm can be larger than one even though the maximum eigenvalue modulus is smaller than one.

Studying a physically meaningful matrix norm instead of the maximum eigenvalue also makes sense from a physical point of view. According to theory, a spectral radius smaller than one is enough to guarantee stability. But strictly speaking, it only says that perturbations are eliminated for $t \to \infty$, and they can very well be amplified in the meantime. As asymptotic stability implies stability, perturbations are bounded but in some cases the bound can be quite large. On the other hand, if one of the above matrix norms is smaller than one, there would be a contraction of perturbations in terms of the chosen norm over each cycle.

To sum up the ideas of this section: two different goals can be pursued by minimizing some 'physical' induced matrix norm:

- Reduce this norm below one to have a contraction of perturbations in terms of this norm. One should be aware that this is a very strict criterion and that it will be hard to reach this goal for many physical systems. In section 5.3 we will discuss a way to soften this criterion.
- Reduce this norm not necessarily as far as one, but use this upper bound only to provoke an overall decrease of the maximum eigenvalue. Reducing an upper bound does of course not mean that in every step the spectral radius itself will also be decreased – the contrary can be the case. One should carefully observe the value of the spectral radius in every iterate because it might attain its best value (of all iterates) before convergence to the minimum of the chosen norm is achieved.

Remark:
In contrast to eigenvalues, general induced matrix norms do depend on the starting point

of the periodic sample, i. e. they are not the same for different intervals $[t_0, t_0 + T]$ and $[t_1, t_1 + T]$ with $t_0 \neq t_1$.

5.2.3 Singular Value Optimization

As we have seen above (equation (5.30)) the maximum singular value of a matrix is equal to its Euclidean norm and therefore an upper bound to the spectral radius.

By definition, the singular values σ_i of a real matrix C are the eigenvalues of the symmetric, positive semi-definite matrix H

$$\sigma_i(C) = \sqrt{\lambda_i(C^T C)} = \sqrt{\lambda_i(H)}, \tag{5.36}$$

i.e. they are always real and positive. In the mapping of the unit sphere into a hyperellipse by means of the matrix C the singular values of C describe the lengths of the semi-axes (Trefethen & Bau [92]).

So choosing the maximum singular value instead of the maximum eigenvalue as optimization criterion for stability optimization

$$\min \sigma_{max}(C(x)) \tag{5.37}$$

has the following consequences:

- the objective function is still non-differentiable at points with multiple maximum eigenvalue, but
- non-differentiabilities have become less numerous and far less serious. Being eigenvalues of of a symmetric matrix, singular values are always well conditioned and Lipschitz. As all singular values are positive, no absolute values have to be taken. In contrast to eigenvalue optimization we are now only facing points of type b), i.e. the non-differentiability is limited to isolated points.

To summarize, singular value optimization is a considerably easier problem than eigenvalue optimization. Among all matrix norms, the Euclidean norm has the advantage of being the most meaningful from an engineering point of view.

5.2.4 Optimization of 1-norm or ∞-norm

The induced 1-norm and ∞-norm of a matrix are its maximum column sum and maximum row sum, respectively (equations (5.29) and (5.28)). The use of one of these two norms as objective function for stability optimization has the advantage to allow a transformation into a standard nonlinear programming problem. We will show here the transformation for the example of the 1-norm, the proceeding for the ∞-norm is analogous.

The original unconstrained non-differentiable min-max optimization problem

$$\min ||C||_1 = \min \left(\max_{j=1,...,n} ||c_j||_1 \right) = \min \left(\max_{j=1,...,n} \left(\sum_{i=1}^{n} |c_{ij}| \right) \right) \tag{5.38}$$

(where c_j is the j-th column vector) can be reformulated as as differentiable constrained optimization problem

$$\min z \tag{5.39}$$

$$s.t. -y_{ij} \leq c_{ij} \leq y_{ij} \qquad \forall i = 1, ..., n,\ j = 1, ..., n \tag{5.40}$$

$$z \geq \sum_i y_{ij} \qquad \forall j = 1, ..., n \tag{5.41}$$

using the auxiliary variables z and y_{ij} with $i = 1, ..., n; j = 1, ..., n$.

The transformation was performed at the cost of $n^2 + 1$ additional variables and $2n^2 + n$ additional equations. The resulting problem (5.39) - (5.41) is a quite large but simple and structured standard nonlinear programming problem.

The advantage of using one of these two norms in stability optimization is that standard nonlinear programming methods can be used for the solution. Due to the sparsity of the problem it would however be favorable to develop faster special purpose algorithms that exploit the structure.

5.3 Study of Matrix Powers

While in the previous section we have only looked at possible replacements for the spectral radius, we here want to question the use of the monodromy matrix C itself. As it turns out, sometimes it might be favorable to look at powers of C instead.

In the previous section we have discussed that demanding a contraction of the 1-, 2-, or ∞-norm over one cycle is a very strict criterion. A softening of this demand can be achieved by only asking for a contraction of the norm over a number p of cycles with $p > 1$.

For $p \to \infty$ there is equivalence between a contraction – or even disappearance, to be exact – of the norm of C^p and the spectral radius of C being smaller than one, as stated in the following theorem:

Theorem 5.3
For an arbitrary matrix $C \in \mathbb{R}^{n \times n}$ and a norm $|| \cdot ||$ holds:

$$\lim_{p \to \infty} ||C^p|| = 0 \Leftrightarrow \rho(C) < 1 \tag{5.42}$$

Note that the same statement can be made for complex matrices.

Proof:
Direction ($\Leftarrow$) can be proven directly. According to theorem 5.2, there is a norm $||\cdot||_\epsilon$ with small ϵ such that

$$||C||_\epsilon = \rho(C) + \epsilon =: \alpha < 1.$$

Since for any induced matrix norm $||AB|| \leq ||A||\,||B||$,

$$||C^p||_\epsilon \leq ||C||_\epsilon^p = \alpha^p.$$

Due to the equivalence of norms (5.35) follows for any other norm $||\cdot||$

$$||C^p|| \leq c_1||C^p||_\epsilon = c_1\alpha^p$$

which tends to zero for $p \to \infty$.

For the other direction ($\Rightarrow$) we need to take two steps. In a first step we show that

$$\lim_{p\to\infty} ||C^p|| = 0 \Rightarrow \lim_{p\to\infty}(\rho(C^p)) = 0$$

which follows directly from the theorem of Hirsch: the spectral radius of C^p is zero if its upper bound in terms of some induced matrix norm is zero.

In a second step we show the following equivalence (although showing ($\Rightarrow$) would be sufficient for the proof):

$$\lim_{p\to\infty}(\rho(C^p)) = 0 \Leftrightarrow \rho(C) < 1.$$

Every square matrix C has a Schur factorization:

$$C = Q \cdot T \cdot Q^T$$

where Q is orthogonal (i.e. $Q^TQ = I$) and T is upper triangular. The eigenvalues of C always appear as diagonal elements d_i of T because C and T are similar and the eigenvalues of triangular matrices are the diagonal elements:

$$Det|T - \lambda I| = 0 \Rightarrow \Pi_{i=0}^n(d_i - \lambda) = 0$$

for all λ of T and C respectively and therefore

$$d_i = \lambda_i \quad i = 1, ..., n$$

With equation (5.3) follows

$$\begin{aligned} C^p &= (Q \cdot T \cdot Q^T)^p \\ &= Q \cdot T \cdot Q^T \cdot Q \cdot T \cdot Q^T \cdot ... \cdot Q \cdot T \cdot Q^T \\ &= Q \cdot T^p \cdot Q^T \end{aligned}$$

T^p is again upper-triangular and has diagonal elements – and therefore eigenvalues – $d_i^p = \lambda_i^p$. Due to the similarity with T^p, C^p also has eigenvalues λ_i^p, i.e.

$$\rho(C^p) = \rho^p(C) \tag{5.43}$$

(Note that in the special case of diagonalizable – i.e. non-defective – matrices, the same property follows more easily:

$$C^p = (X\Lambda X^{-1})^p = X \cdot \Lambda^p \cdot X^{-1}.)$$

Since

$$\lim_{p\to\infty}(\rho^p(C)) = 0 \Leftrightarrow \rho(C) < 1.$$

this completes the second part of the proof. □

Note however, that induced matrix norms do not exhibit a property similar to (5.43) concerning their powers:

$$||C||^p \neq ||C^p|| \tag{5.44}$$

such that specifically for the maximum singular value

$$\sigma_{max}(C^p) \neq \sigma^p_{max}(C). \tag{5.45}$$

This results from the fact that singular values or induced matrix norms in general are not derived from a similarity transformation such as eigenvalues, and therefore transformation matrices are not eliminated when taking the power of a matrix.

In chapters 7 – 9 we show several plots of matrix norms as functions of matrix powers confirming this statement.

One has to be aware, that if the spectral radius of the monodromy matrix is larger than one, the spectral radius and induced matrix norms for increasing powers of the matrix will be highly divergent.

To sum up, studying powers of a matrix can be a good alternative to studying the matrix itself. From the theoretical point of view it does not make a difference which specific norm (including the spectral radius) is chosen as optimization criterion as long as the matrix power is high enough. What is a good choice for the power p in a practical case needs to be heuristically determined.

5.4 Summary: Objective Functions for Stability Optimization

We finally summarize possible objective functions for stability optimization that will be evaluated and compared in this study:

- the original eigenvalue criterion:

$$f_1 = |\lambda_{max}(C)| = \rho(C) \tag{5.46}$$

- induced matrix norms of the monodromy matrix:

$$f_{2a} = \sigma_{max}(C) = ||C||_2 \tag{5.47}$$
$$f_{2b} = ||C||_\infty \tag{5.48}$$

- induced matrix norms of a power of the monodromy matrix:

$$f_{3a} = \sigma_{max}(C^p) = ||C^p||_2 \tag{5.49}$$
$$f_{3b} = ||C^p||_\infty \tag{5.50}$$

The advantages of one choice of norm over the other as well as a good choice of the matrix power will also be investigated.

Chapter 6

Numerical Methods for Stability Optimization

This chapter contains a collection of useful methods for stability optimization problems as formulated in section 5.1.3. Since not all these methods were necessary to find solutions for the application of robot stabilization, some (like the derivatives of singular values) have so far only been implemented as library modules. The evaluation of different possible methods for stability optimization was possible in the framework of an object-oriented optimization library that we developed during the research for this thesis (see appendix).

We briefly review literature about general non-smooth optimization and eigenvalue optimization in section 6.1. Section 6.2 describes the specific direct search method, a variant of the Nelder-Mead polytope algorithm, that we have implemented and used for stability optimization of all our robot examples. In section 6.3 we give apparently new formulas for the computations of derivatives of simple and multiple singular values. Section 6.4 recalls the computation of the monodromy matrix in the presence of discontinuities. The necessary projections for monodromy matrices of autonomous systems are described in section 6.5. In section 6.6 we present, for the first time, formulas for the derivatives of monodromy matrices for discontinuous dynamic equations with respect to initial values and parameters. They represent second order derivatives of the differential equations. Section 6.7 finally contains a numerical procedure for the determination of nonlinear stability margins.

6.1 Review of Literature

6.1.1 Non-differentiable Optimization

All smooth optimization methods are directly or indirectly based on a Taylor series approximation of the objective function and require at least continuous differentiability of

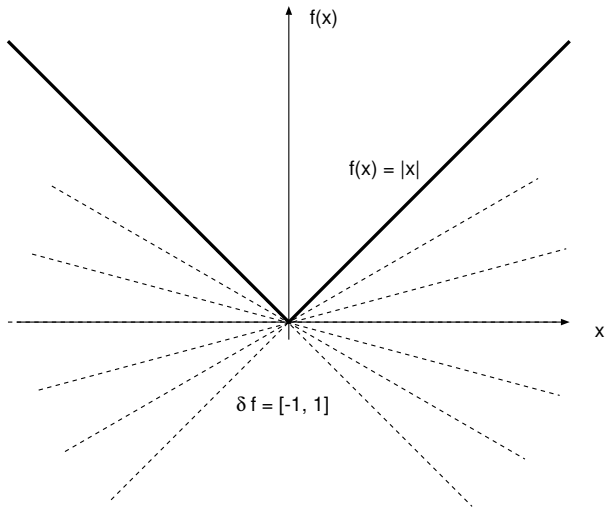


Figure 6.1: Subgradients at $x = 0$ for the example function $|x|$

this function. In the case of a non-differentiable objective functions like the analyzed case of eigenvalue optimization or non-differentiabilities introduced by the method, like a non-smooth penalty function, smooth optimization methods can not be applied in general. We summarize in this section some important theoretical background information about non-differentiable optimization. More details can be found in Clarke [16] or in the respective chapters in Großmann & Terno [36] and Fletcher [28].

In the case of non-differentiable functions, substitutes for the derivative are required. For convex functions, the subdifferential is defined as

$$\partial f_s(x) = \{s \in \mathbb{R}^n : f(y) \geq f(x) + s^T(y - x) \; \forall \; y \in \mathbb{R}^n\} \tag{6.1}$$

If f attains only finite values, ∂f_s is a nonempty, compact convex set. The elements of ∂f_s are called the subgradients. Figure 6.1 shows the subdifferential for the example function $|x|$.

The generalized gradient of Clarke [16] is defined for a more general class of functions. Here only Lipschitz continuity of f is required. Then the generalized directional derivative of f, evaluated in the direction d is given by

$$d\, f(x; d) = \lim_{y \to x,\, h \to 0+} \sup \frac{1}{h} \left(f(y + dh) - f(y)\right). \tag{6.2}$$

The generalized gradient is defined as

$$\partial f(x) = \{\xi \in \mathbb{R}^n : df(x, d) \geq d \cdot \xi \; \forall \; d \in \mathbb{R}^n\}. \tag{6.3}$$

The first order necessary optimality condition for for non-smooth optimization problems is

$$0 \in \partial f(x^*) \tag{6.4}$$

which is a generalization of $\nabla f(x^*) = 0$ for smooth optimization problems.

For general non-smooth optimization problems special optimization techniques are required. The gradient-based algorithms can be divided into three basic types:

- Subgradient methods:
  They form the equivalent of the steepest descent method for smooth nonlinear optimization. The iterates are computed as

$$x^{k+1} = x^k + \alpha_k d^k \tag{6.5}$$

  with

$$d_k = \frac{-s_k}{||s_k||} \quad \text{for some } s_k \in \partial f(x^k), \tag{6.6}$$

  It is not advisable to determine the steplength α_k by a line search, as cases of convergence towards a non-optimal point can simply be constructed (Fletcher [28]). Instead, a priori fixed steplengths α_k satisfying the conditions

$$\sum_{k=0}^{\infty} \alpha_k = \infty \tag{6.7}$$

$$\sum_{k=0}^{\infty} \alpha_k^2 < \infty \tag{6.8}$$

  should be used. In this case, (very slow) convergence can be guaranteed, but there is of course no assured improvement in every step.

- Bundle methods:
  The idea of bundle methods is similar to that of conjugate gradient methods for nonlinear optimization. The direction of search is determined by bundled subgradient or generalized gradient information of the current iterate and previous ones. In the simplest form of the algorithm, the initial bundle is set to $s^{(0)} \in \partial f(x^{(0)}$, and subgradients $s^{(k)} \in \partial f(x^{(k)}$ are added in successive iterations. A reset of the bundle is performed from time to time. For some choices of the bundle convergence of the algorithm can be proven.

- Parameterized embedding in smooth problems:
  The non-smooth objective function is substituted by a parameterized smooth objective function that degenerates to the original non-smooth function in the limit case $\epsilon \to 0$. In analogy to penalty function techniques, a whole family of auxiliary problems with different parameter values ϵ is generated and solved. Appropriate control of the parameter ϵ is required.

Non-smooth optimization literature (e.g. Fletcher [28]) very often focuses on the special case of composite non-smooth optimization problems

$$f(x) = h(c(x)) \tag{6.9}$$

where $h(c)$ is convex, but non-smooth and $c(x)$ is a vector of smooth functions. For this case, more methods, like e.g. a non-smooth extension of the SQP method have been developed. However, min-max eigenvalue optimization does not belong to this type of problems.

Since the maximum eigenvalue of a non-symmetric matrix is neither differentiable nor Lipschitz (see section 5.1.1), both the theory of subgradients and Clarke's theory of generalized gradients are not applicable in this case.

A very interesting alternative for general non-smooth problems are direct search methods which, in contrast to the methods discussed above, do not require any gradient-like information. Since we have chosen to use an algorithm of this type for stability optimization, these methods are treated more extensively in section 6.2.

6.1.2 Existing Methods for Simpler Cases of Eigenvalue Optimization

The purpose of this section is to refer to some important literature in the field of eigenvalue optimization. Most articles we have found concentrate on the case of symmetric affine matrix functions $A(x)$ which has the advantage of leading to convex objective functions $|\lambda_{max}(x)|$ The definition of a subdifferential is possible in this case.

There is a series of publications by Overton. A quadratically convergent algorithm for symmetric affine matrices was proposed in Overton [68] and extended to large scale matrices in Overton [69]. Shapiro & Fan [83] and Overton & Womersley [71] give the corresponding second order convergence analysis. Goh & Teo [33] have attempted a solution of eigenvalue optimization problems by min-max reformulation of eigenvalue optimization and application of standard algorithms, but other sources (see e.g. Panier [72]) give examples for a failure of this approach. Eigenvalue optimization problems for symmetric matrices can be transformed into semidefinite programming problems which have recently received a lot of attention. There is extensive literature on this subject (see e.g. Alizadeh et al. [2] for primal-dual interior point methods and Helmberg & Rendl [39] for spectral bundle methods).

The only publication about eigenvalue optimization for non-symmetric matrices that we are aware of is Overton & Womersley [70]. For affine matrix functions they derive first order optimality conditions and formulas for the direction of descent in the case of non-defective multiple eigenvalues. The defective case is only briefly discussed. The results of this paper do not apply to cases with nonlinear matrix functions or applications involving constraints.

We can summarize that to our knowledge the specific form of eigenvalue optimization problem that we are facing in our study is not addressed in eigenvalue optimization literature.

6.2 A Direct Search Method for Eigenvalue Optimization

Direct search methods are optimization methods that solely use function information and do neither compute nor explicitely approximate derivatives. While derivative-based methods are extremely efficient for many types of problems they reach their limitations if objective function are non-smooth by nature or their values can only inaccurately be determined, or when sensitivity information is not available or at least not reliable. Standard optimization literature (Gill et al. [32]) recommends direct search methods as methods of choice for non-differentiable optimization.

For the solution of the two-level stability optimization problem a direct search method has proven to be a good choice. We use a modification of the direct search method of Nelder & Mead, also known as polytope algorithm. We have used this algorithm not only for eigenvalue optimization but also for minimization of the other criteria listed in section 5.4 since we did not want to blur the comparison of different objective functions by algorithmic influence.

In section 6.2.1 the original algorithm (Nelder & Mead [66], Gill et al. [32]) is presented. Section 6.2.2 describes the modifications we have introduced in order to make the algorithm suitable for stability optimization. We conclude with a short discussion of convergence properties of the Nelder-Mead algorithm and related methods in section 6.2.3. Since we apply the algorithm for parameter optimization, we call the vector of independent variables p throughout this section.

6.2.1 Original Polytope Algorithm

The Nelder-Mead algorithm dating from 1965 is among the most famous optimization methods ever created and is still popular today. Nelder & Mead call the algorithm a simplex method since it is based on a sequence of $(n+1)$-vertex simplices for optimization n-dimensional space $p \in \mathbb{R}^n$ – however, the method is not to be confused with the even older and more famous simplex method for linear programming by Dantzig. This simplex – or polytope – retains information about function values at $n+1$ distinct points and thus obtains some sort of coarse grid sensitivity information. Always replacing its worst point and adapting its shape to the topology the polytope wanders through the space of optimization variables towards a minimum.

The individual steps of one algorithm iteration are the following (also compare figure 6.2):

1. At the beginning of every iteration k, the function values $f(p_i)$ of all $n+1$ vertices are determined, and the vertices are ordered and labeled $p_1, ...p_{n+1}$ such that

$$f(p_1) \leq f(p_2) \leq ... \leq f(p_{n+1}) \tag{6.10}$$

   Vertex p_1 therefore represents the best point of the function known so far. In

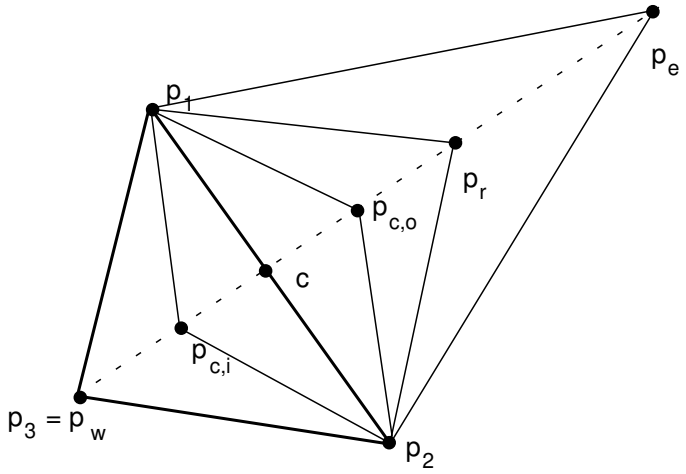


Figure 6.2: Basic concept of polytope algorithm

practice, only the best, worst and second-worst points have to be isolated, but the above fully ordered form is nicer for presentation.

2. The centroid of all vertices but the worst is computed

$$c = \sum_{i=1}^{n} p_i \tag{6.11}$$

The worst vertex $p_w = p_{n+1}$ is reflected on this centroid

$$p_r = c + \alpha(c - p_{n+1}) \tag{6.12}$$

with a reflection coefficient α, $(0 < \alpha < 1)$. If the function value at this reflected point $f(p_r)$ is

$$f(p_1) \leq f(p_r) < f(p_n), \tag{6.13}$$

i.e. neither the new best nor worst point, point p_r replaces p_{n+1} and the iteration step terminates.

3. If

$$f(p_r) < f(p_1), \tag{6.14}$$

the reflection has produced a new minimum and it might be worthwhile trying to go further in this promising direction. This motivates the expansion step

$$p_e = c + \gamma(p_r - c) = \gamma p_r + (1 - \gamma)c \tag{6.15}$$

with expansion coefficient γ, $(\gamma > 1)$. If $f(p_e) < f(p_r)$, the expanded point is accepted as new simplex point, otherwise p_r is accepted. The iteration step is terminated.

4. If on the contrary, after reflection

$$f(p_r) > f(p_n), \tag{6.16}$$

i.e. the reflected point is still the worst point, a contraction of the polytope is performed. Of the two points p_r or p_{n+1} the one with the better function value is chosen as target for the contraction which is accordingly called outer contraction

$$p_{c,o} = c + \beta(p_r - c) = \beta p_r + (1 - \beta)c \tag{6.17}$$

or inner contraction

$$p_{c,i} = c + \beta(p_{n+1} - c) = \beta p_{n+1} + (1 - \beta)c \tag{6.18}$$

with contraction coefficient β, $(\beta < 1)$. If $f(p_c) < f(p_a)$ with $p_a = min(p_r, p_{n+1})$, the contracted point is accepted as new simplex point and the step terminates.

5. For a failed contraction the full polytope is shrunk towards the best point, i.e. the points $p_2, ..., p_{n+1}$ are replaced by

$$x_i' = 0.5(p_i + p_1) \tag{6.19}$$

and a new iteration step starts.

The reflection, expansion and contraction coefficients are heuristically chosen; a frequent choice is $\alpha = \beta = 0.5, \gamma = 2$.

The iteration terminates when the difference of function values of all simplex points in terms of a 'standard error' falls below a chosen tolerance

$$\sqrt{\sum_{i=1}^{n+1}(f(p_i) - f(c))} < tol_t. \tag{6.20}$$

In order to include constraints on the volume to be searched, the authors propose to use a modified objective function like the logarithm of the original function in order to exclude negative values of variables. Linear equality constraints can of course be handled by explicit elimination of variables and reduction of the simplex dimension.

6.2.2 Overview of Necessary Modifications for Stability Optimization

For a better performance in the context of stability optimization we have applied a number of modifications to the original algorithm.

Multiple Expansions

Instead of a single expansion like in the original algorithm we allow multiple expansions if the first expansion was successful. In every step the expansion coefficient γ in equation (6.15) is augmented (e.g. doubled) until there is no further improvement of the objective function. The maximum number of expansions can be specified by the user (default value = 5). In our examples considerable reductions of the objective function have been achieved by these multiple expansion steps.

Modified Contractions

We modified the contraction procedure such that the reflected point (outer contraction) or the original worst point (inner contraction) is contracted towards the best point instead of the centroid. In equations (6.17) and (6.18) c is replaced by p_1.

Modified Shrinking

Since we have observed that frequent polytope shrinking leads to premature convergence we instead allow multiple polytope contractions before shrinking the polytope. The maximum number of contractions can be modified by the user (default value = 5).

Initial Polytope Scaling

Different orders of magnitude of the optimization variables are considered in the choice of the original polytope by an appropriate scaling of the polytope side lengths. Starting from one initial point p_1 provided by the user, the other n polytope points are created by scaled steps in one variable direction each:

$$p_j = p_1 + l_0 \cdot s_{j-1} \cdot e_{j-1}, \quad j = 2, ..., n+1 \tag{6.21}$$

where l_o denotes the unity side length, s_i the scaling factor, and e_i the i-th unity vector in $\mathbb{R}^n$. Scaling factors as well as unity side length can be specified by the user.

Handling of Box Constraints

Box constraints to the parameter space are not handled by a modification of the objective function but by a modification of the algorithm itself.

The constraints have to be considered during reflection, expansion, and initialization steps. Figure 6.3 illustrates a modified reflection step in the presence of box constraints. If a reflected point turns out to be outside the bounds it is set back onto the bounds by the algorithm. The same is done during polytope initialization. If the initial polytope risks to

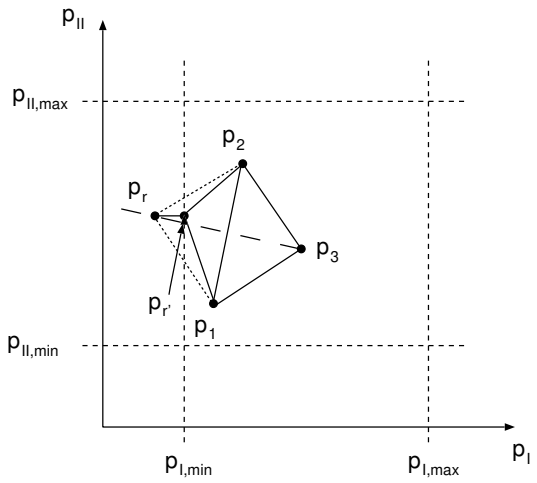


Figure 6.3: Handling of box constraints with modified polytope algorithm

degenerate due to multiple resets the user will be prompted to modify the initial polytope. Directions of variable space in which a reset had to be performed are excluded from the expansion steps. This leads to a direction of expansion different from the direction of reflection. The necessity for a reset is checked after each expansion step.

Due to the convex nature of box constraints no modifications have to be made to contraction and shrinking procedures and to the computation of the centroid.

Termination Criterion

If the polytope algorithm is applied to eigenvalue optimization, ill-conditioning of the objective function may occur. This is especially true for the optimum were typically the maximum eigenvalue is multiple. It is therefore sometimes favorable to increase the termination tolerance in the presence of large eigenvalue condition numbers.

Restart Procedure

For a problem with multiple minima like stability optimization we wanted to reduce the danger of converging to some local minimum if there is another one of better function value nearby. After convergence we therefore perform a restart by generating a new initial polytope keeping the optimum as one point of the new polytope. A minimum is only accepted if convergence to the same point has appeared twice. For our computations this restart procedure typically causes a considerably improved objective function value at final convergence relative to the first convergence point.

6.2.3 Discussion of Convergence Properties

Algorithms of the Nelder-Mead simplex type have proven to be very robust for many applications although not many theoretical results have been obtained so far. Until recently there was no theoretical analysis explicitely treating the original algorithm but only variants. There are two recent publications studying convergence in low dimension. Lagarias et al.[46] gave a proof of convergence of the polytope algorithm for one-dimensional and certain two-dimensional functions. McKinnon [55], however, demonstrated convergence to non-stationary points for another class of two-dimensional functions. No proofs have been presented so far for functions of higher dimensions. There are reports about the polytope degenerating in high dimensions causing the algorithm to fail.

There is a class of direct search methods, called pattern search algorithms, for which more theoretical results are available. Instead of replacing only the worst point like the polytope algorithm, pattern search algorithms replace all but the best point. Torczon [91], [90] proved that pattern search algorithms converge to a stationary point when applied to smooth functions. But since the smoothness condition is not satisfied by eigenvalue optimization problems and since n function evaluations are required for one step of the pattern search methods, we have favored our variant of the Nelder-Mead algorithm over pattern search.

We are aware that convergence of the algorithm cannot be proven but have been motivated by the fact that it converges to the optimum for a number of standard NLP test problems.

6.3 Numerical Methods for Singular Value Optimization

As we have outlined in the previous chapter, singular value optimization represents an interesting alternative to eigenvalue optimization in the intention to increase a system's stability.

Numerical libraries like LAPACK (see Anderson et al. [3]) contain very efficient routines for the computation of singular values such that there is no need for development of new methods. In this section we concentrate on formulas for the derivatives of the maximum singular value with respect to an independent optimization variable x_k on which the matrix C implicitly depends:

$$\frac{d\sigma_{max}}{dx_k} = \frac{d\sigma_{max}}{dC} \circ \frac{dC}{dx_k} = \left\langle \frac{d\sigma_{max}}{dC}, \frac{dC}{dx_k} \right\rangle . \tag{6.22}$$

x_k is not to be confused with the vector of state variables x used in previous chapters. $\langle \cdot \rangle$ denotes the Frobenius product or inner matrix product

$$\langle A, B \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{ij}. \tag{6.23}$$

We distinguish the cases of simple and multiple maximum singular value.

6.3.1 Computation of Derivatives if Singular Value is Simple

We assume that singular values have sorted order after computation by the library routine and that the maximum value is simple:

$$\sigma_{max} = \sigma_1 > \sigma_2 \geq ..... \geq \sigma_n. \tag{6.24}$$

In this case the maximum singular value is continuously differentiable. The computation of the derivative does not pose a big problem but we will see that there is nevertheless some room for improvement.

Let us recall from literature (e.g. Trefethen & Bau [92]) that there are two different ways to compute the singular values of a matrix C.

The first variant is to use the formal definition

$$\sigma_i(C) = \lambda_i(\sqrt{C^T C}), \quad C \in \mathbb{R}^{n \times n} \tag{6.25}$$

where $B = C^T C$ is the covariance matrix of C. It has the disadvantage of a worse condition than the original matrix as the errors are squared. Furthermore it is numerically unstable.

An alternative computation of singular values is based on a symmetric auxiliary matrix H of double dimension:

$$H = \begin{pmatrix} 0 & C^T \\ C & 0 \end{pmatrix}, \quad H \in \mathbb{R}^{2n \times 2n}. \tag{6.26}$$

The condition number of H is the same as for the original matrix C. H has got $2n$ eigenvalues which come in pairs of opposite sign. The singular values of C are equal to the positive eigenvalues (or the absolute values of the eigenvalues) of H;

$$\lambda_i(H)^{+/-} = \pm\, \sigma_i(C) \tag{6.27}$$

The corresponding $2n$ eigenvectors $v_{ev_i}^{+/-}$ of H are related to the left and right singular vectors u_i and v_i of C by

$$v_{ev_i}^{+/-} = \frac{1}{\sqrt{2}} \begin{pmatrix} v_i \\ \pm u_i \end{pmatrix}. \tag{6.28}$$

It is this second variant that we are going to use for the computation of derivatives although we will try to avoid the computation of matrices of dimension larger then n.

Using formula (5.8) for the derivatives of eigenvalues of symmetric matrices presented in the previous chapter and relation (6.28) for the eigenvectors of H, we can conclude for the derivatives of singular values:

$$\frac{d\sigma_1(C)}{dx_k} = \frac{d\lambda_1^+(H)}{dx_k}$$

$$
\begin{aligned}
&= \left\langle \frac{d\lambda_1^+}{dH}, \frac{dH}{dx_k} \right\rangle \\
&= \left\langle \frac{1}{2} \begin{pmatrix} v_1 \\ u_1 \end{pmatrix} \begin{pmatrix} v_1 \\ u_1 \end{pmatrix}^T, \begin{pmatrix} 0 & \frac{dC}{dx_k}^T \\ \frac{dC}{dx_k} & 0 \end{pmatrix} \right\rangle \\
&= \frac{1}{2} \left\langle \begin{pmatrix} v_1 v_1^T & v_1 u_1^T \\ u_1 v_1^T & u_1 u_1^T \end{pmatrix}, \begin{pmatrix} 0 & \frac{dC}{dx_k}^T \\ \frac{dC}{dx_k} & 0 \end{pmatrix} \right\rangle \\
&= \frac{1}{2} \left(\left\langle v_1 u_1^T, \frac{dC}{dx_k}^T \right\rangle + \left\langle u_1 v_1^T, \frac{dC}{dx_k} \right\rangle \right) \\
\frac{d\sigma_1(C)}{dx_k} &= \left\langle u_1 v_1^T, \frac{dC}{dx_k} \right\rangle
\end{aligned} \tag{6.29}
$$

We have thus established a formula for the derivative of a (simple) singular value with respect to an independent variable x_k that only requires computation and multiplication of two square matrices of dimension n but does not have the ill-conditioning drawback of equation (6.25). We will come back to the second part of equation (6.29), $\frac{dC}{dx_k}$, in section 6.6.

6.3.2 Computation of Derivatives if Singular Value is Multiple

The case of a maximum singular value of multiplicity m

$$\sigma_{max} = \sigma_1 = \sigma_2 = ... = \sigma_m > \sigma_{m+1} \geq ..... \geq \sigma_n \tag{6.30}$$

is more complicated. As we have discussed in section 5.2.3, the individual singular values are non-differentiable but well-conditioned at the points of multiple maximum singular value. Two tasks have to be handled:

- detection of a local minimum
- computation of a direction of descent.

Both can be solved based on the idea, that the derivatives of the individual singular values at this non-defined point can be approximated by the corresponding derivatives at a slightly perturbed point.

$$\frac{d\sigma_i(C(x))}{dx_k} \approx \frac{d\sigma_i(C(x+x')}{dx_k} \tag{6.31}$$

We apply a perturbation to the matrix C that splits the multiple singular value into m distinct ones, i.e. we apply a different perturbations in the direction of each of the multiple singular values . From the singular value decomposition of matrix C

$$C = U\Sigma V^T \tag{6.32}$$

we create the perturbed matrix

$$C_{pert} = U(\Sigma + \Sigma')V^T = C + U\Sigma' V^T \tag{6.33}$$

with the diagonal matrix of perturbations

$$\Sigma' = \begin{pmatrix} m-1 & & & & & 0 & \cdots & 0 \\ & m-2 & & & & 0 & \cdots & 0 \\ & & \ldots & & & 0 & \cdots & 0 \\ & & & 1 & & 0 & \cdots & 0 \\ & & & & 0 & 0 & \cdots & 0 \\ 0 & & \cdots & & 0 & 0 & \cdots & 0 \\ \cdots & & \cdots & & & & \cdots & \\ 0 & & \cdots & & 0 & 0 & \cdots & 0 \end{pmatrix} \cdot \epsilon \tag{6.34}$$

where ϵ is some constant, e.g. $\epsilon = 10^{-6}$

For this perturbed matrix which has a simple maximum singular value the derivative can be computed as described in the last section.

In order to detect a possible local minimum at points of multiple maximum singular value, we have to verify if the subgradient $\partial f = 0$ is element of the subdifferential at this point. In practice, this results in the need to check if for all directions k of the optimization space there are two gradients of individual singular values which are of opposite signs (or one that is zero):

$$\frac{d\sigma_i(C(x))}{dx_k} \cdot \frac{d\sigma_j(C(x))}{dx_k} \leq 0. \tag{6.35}$$

6.4 Computation of Monodromy Matrices for Discontinuous Differential Equations

In this section, we discuss the numerical computation of the monodromy matrix associated with a solution of the inner loop discontinuous periodic optimal control problem. Let us recall that the monodromy matrix is equivalent with the Jacobian of the Poincaré map of the periodic solution

$$C_x = \frac{dx(T)}{dx(0)} = C_{q,v} \tag{6.36}$$

with x being again the vector of state variables $x^T = (q^T, v^T)$.

For the optimal control problem solution, we have already computed the sensitivities of integration end values with respect to initial values on each multiple shooting interval $C_x(t_{i-1}, t_i)$ and can simply reuse them at this point. For continuous model equations and state variables the monodromy matrix over the whole period would be produced by a chain rule multiplication of the individual sensitivity matrices:

$$C_x(0,T) = C_x(t_0, t_m) = C_x(t_{m-1}, t_m) \cdot \ldots \cdot C_x(t_1, t_2) \cdot C_x(t_0, t_1). \tag{6.37}$$

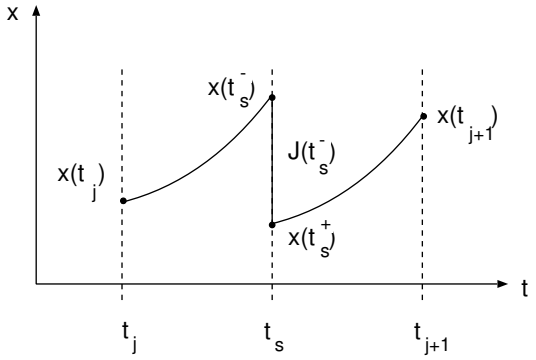


Figure 6.4: Implicitly defined discontinuity of state variables at t_s

or

$$C_{q,v}(0,T) = C_{q,v}(t_{m-1},t_m) \cdot \ldots \cdot C_{q,v}(t_1,t_2) \cdot C_{q,v}(t_0,t_1). \tag{6.38}$$

As our dynamic models contain discontinuities of state variables and/or the right hand side depending implicitly on q, v, and t, any perturbation will cause the discontinuities to occur sooner or later than for the original solution. In this case, the sensitivity information has to be updated at the discrete point of time t_s of the discontinuity (Bock [10], von Schwerin et al. [96]).

We include here a full derivation of the update formula since it illustrates the principles that we will also need in section 6.6 for computing the derivatives of the monodromy matrix. The discontinuity is situated at t_s in the interval $[t_j, t_{j+1}]$ (compare figure 6.4). The switching function is given as

$$s(t_s,x,p) = s(t_s,q,v,p) \tag{6.39}$$

with partial derivatives s_t and $s_x = s_{q,v}$. Discontinuities are described in terms of right hand side changes $f(t_s^+) - f(t_s^-)$ and of state variable jump functions

$$J(t_s,x,p) = J(t_s,q,v,p) = \begin{pmatrix} q^+(t_s) - q^-(t_s) \\ v^+(t_s) - v^-(t_s) \end{pmatrix} \tag{6.40}$$

with partial derivatives J_t and $J_x = J_{q,v}$.

We look for the derivative

$$\frac{dx(t_{j+1})}{dx(t_j)} \tag{6.41}$$

where the dependencies of $x(t_{j+1})$ and $x(t_s^+)$ are precisely stated as

$$x(t_{j+1}; t_s(x_j,p), x(t_s^+), p) \tag{6.42}$$

and

$$\begin{aligned} x(t_s^+) &= x(t_s^-) + J(t_s, x(t_s^-), p) \\ &= x(t_s^-(x_j,p), x_j, p) + J(t_s^-(x_j,p), x(t_s^-(x_j,p), x_j, p), p). \end{aligned} \tag{6.43}$$

This leads to

$$\frac{dx(t_{j+1})}{dx(t_j)} = \underbrace{\frac{\partial x(t_{j+1})}{\partial t_s}}_{=:\mathcal{A}} \underbrace{\frac{dt_s}{dx_j}}_{=:\mathcal{B}} + \frac{\partial x(t_{j+1})}{\partial x(t_s^+)} \underbrace{\frac{dx(t_s^+)}{dx_j}}_{=:\mathcal{C}} \tag{6.44}$$

where the three terms $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ remain to be determined.

$\mathcal{A}$ follows from standard Analysis

$$\frac{\partial x(t_j+1)}{\partial t_s^+} = -\frac{\partial x_{j+1}}{\partial x(t_s^+)} \cdot \dot{x}(t_s^+). \tag{6.45}$$

$\mathcal{B}$ can be computed using the implicit function theorem

$$s(t_s(x_j,p), x(t_s(x_j,p), x_j, p), p) = 0 \Rightarrow \tag{6.46}$$

$$\frac{ds}{dx_j} = s_t(t_s^-)\frac{dt_s}{dx_j} + s_x(t_s^-)\dot{x}(t_s^-)\frac{dt_s}{dx_j} + s_x(t_s^-)\frac{dx(t_s^-)}{dx_j} = 0 \tag{6.47}$$

$$\underbrace{\left(s_t(t_s^-) + s_x(t_s^-)\dot{x}(t_s^-)\right)}_{=:\dot{s}(t_s^-)}\frac{dt_s}{dx_j} + s_x(t_s^-)\frac{\partial x(t_s^-)}{\partial dx_j} = 0 \tag{6.48}$$

$$\mathcal{B} = \frac{dt_s}{dx_j} = -\frac{-s_x(t_s^-)^T}{\dot{s}(t_s^-)} \cdot \frac{\partial x(t_s^-)}{\partial x_j} \tag{6.49}$$

$\mathcal{C}$ follows from strict derivation of (6.43):

$$\begin{aligned} \mathcal{C} &= \frac{dx(t_s^+)}{dx_j} = \dot{x}(t_s^-)\frac{dt_s}{dx_j} + \frac{\partial x(t_s^-)}{\partial x_j} + J_t\frac{dt_s}{dx_j} + J_x\left(\dot{x}(t_s^-)\frac{dt_s}{dx_j} + \frac{\partial x(t_s^-)}{\partial x_j}\right) \\ &= \left(\dot{x}(t_s^-) + J_t + J_x\dot{x}(t_s^-)\right)\cdot\frac{dt_s}{dx_j} + (I + J_x)\cdot\frac{\partial x(t_s^-)}{\partial x_j} \end{aligned} \tag{6.50}$$

where again we need to substitute term $\mathcal{B}$.

We thus finally obtain

$$\begin{aligned} \frac{dx(t_{j+1})}{dx(t_j)} &= \frac{dx(t_{j+1})}{dx(t_s^+)} \cdot \left(\left(\dot{x}(t_s^+) - \dot{x}(t_s^-) - J_x\dot{x}(t_s^-) - J_t\right)\cdot\frac{s_x^T}{\dot{s}} + I + J_x\right)\cdot\frac{dx(t_s^-)}{dx(t_j)} \\ &= C_x(t_s^+, t_{j+1})\cdot U_x \cdot C_x(t_j, t_s^-). \end{aligned} \tag{6.51}$$

We can conclude that state-dependent discontinuities require the inclusion of an update term U_x in the chain rule multiplication of equation (6.37) for the computation of the monodromy matrix with

$$U_x = \left(f(t_s^+) - f(t_s^-) - J_x f(t_s^-) - J_t\right)\cdot\frac{s_x^T}{\dot{s}} + I + J_x \tag{6.52}$$

or, using position and velocity variables

$$U_{q,v} = \left(f(t_s^+) - f(t_s^-) - J_t - J_{q,v}f(t_s^-)\right)\cdot\frac{1}{\dot{s}}(s_q^T, s_v^T) + I + J_{q,v}. \tag{6.53}$$

6.5 Projection of Monodromy Matrix for Autonomous Systems

According to theorem 4.4, the monodromy matrix of autonomous systems has always one eigenvalue equal to one. It does not make sense to optimize the maximum eigenvalue of this original matrix or some upper bound to this eigenvalue as for the case where all other eigenvalues are smaller than one, one would senselessly be pushing on this invariant eigenvalue of one.

Instead, a projection needs to be performed in order to eliminate the direction of this eigenvalue from the monodromy matrix before optimization. Recalling section 3.1 about dynamical systems, this is equivalent with studying the Jacobian of a Poincaré map that is not produced by regular time strobes (figure 3.1a) but by the intersection with some $(n-1)$-dimensional manifold in state space (figure 3.1b).

A projection matrix has to be chosen such that the eigenvalue of one is eliminated but all other eigenvalues of the matrix are conserved. This requirement is fulfilled by the rank(n-1) orthogonal projector

$$P_{\perp q_1} = I - q_1 q_1^T \tag{6.54}$$

with q_1 being the normalized right eigenvector associated with $\lambda = 1$. Note that for an orthogonal projector we have $P = P^T$, but not $P^T P = I$ as an orthogonal projector is not equivalent to an orthogonal matrix.

In order to obtain the projected matrix we need to project both rows and columns onto this subspace:

$$C_{proj} = P_{\perp q_1} \cdot C P_{\perp q_1}. \tag{6.55}$$

In the case where a power of the matrix instead of the matrix itself is studied (compare section 5.3), the projection is performed after multiplication:

$$C^p_{proj} = P_{\perp q_1} \cdot C^p P_{\perp q_1} \tag{6.56}$$

Note that the monodromy matrix C of an autonomous system and its p-th power C^p both have an eigenvalue of one with the same associated eigenvectors. They therefore result in the same projection matrix $P_{\perp q_1}$.

The derivative of a projected matrix with respect to the k-th component of the state variable vector x is computed as:

$$\begin{aligned} \frac{dC_{proj}}{dx_k} &= \frac{d}{dx_k}(P_{\perp q_1} \cdot C \cdot P_{\perp q_1}) \\ &= P_{\perp q_1} \cdot \frac{dC}{dx_k} \cdot P_{\perp q_1}. \end{aligned} \tag{6.57}$$

6.6 Computation of First Order Derivatives of the Monodromy Matrix

For a derivative-based method for stability optimization, one would also require derivatives of the monodromy matrix with respect to initial values and parameters. They represent second order derivatives of the trajectories. In this section we derive formulas for the derivatives of the monodromy matrix in a multi-interval context such as multiple shooting assuming that the derivatives on the individual intervals have been determined. We distinguish the cases of continuous and discontinuous differential equations. We are not aware of any reference having stated these formulas before. As the full derivatives of the monodromy matrix represent three-dimensional tensors, we prefer to give instead an expression for the k-th component which is a matrix.

6.6.1 Continuous Dynamics

First we concentrate on the simpler case with no discontinuities in the dynamical equations. It is however not as straightforward as for first order derivatives since for second order derivatives there is no simple chain rule dependency similar to equation (6.37).

Picture two consecutive intervals $[t_j, t_{j+1}]$ and $[t_{j+1}, t_{j+2}]$ for which the dependencies of the respective integration end values are fully described by

$$x(t_{j+1}) = x(t_{j+1}; t_j, x_j, p) \tag{6.58}$$
$$x(t_{j+2}) = x(t_{j+2}; t_{j+1}, x(t_{j+1}; t_j, x_j, p), p). \tag{6.59}$$

We will give equations for the derivatives of the monodromy matrix C_x with respect to initial values x_j and parameters p. Throughout this section we will use the abbreviation $x_i = x(t_i)$.

The first and second derivatives of the integration end values with respect to initial values and parameters on the two intervals are assumed to be known:

$$\begin{aligned}
C_x(t_j, t_{j+1}) &= \frac{dx_{j+1}}{dx_j} & C_x(t_{j+1}, t_{j+2}) &= \frac{dx_{j+2}}{dx_{j+1}} \\
C_p(t_j, t_{j+1}) &= \frac{dx_{j+1}}{dp} & C_p(t_{j+1}, t_{j+2}) &= \frac{dx_{j+2}}{dp} \\
D_x(t_j, t_{j+1}) &= \frac{d^2x_{j+1}}{dx_j^2} & D_x(t_{j+1}, t_{j+2}) &= \frac{d^2x_{j+2}}{dx_{j+1}^2} \\
D_p(t_j, t_{j+1}) &= \frac{d^2x_{j+1}}{dx_j dp} & D_p(t_{j+1}, t_{j+2}) &= \frac{d^2x_{j+2}}{dx_{j+1} dp}.
\end{aligned}$$

By $D_{x,k}$ and $D_{p,k}$ we denote the k-th matrix components of the derivatives D_x and D_P and by $C_{x,k}$ and $C_{p,k}$ the k-th row vector of some matrix C_x or C_p respectively.

Derivative with Respect to Initial Values

We look for the k-th derivative of the overall monodromy matrix $C_x(t_j, t_{j+2})$ with respect to initial values:

$$\begin{aligned} D_{x,k}(t_j, t_{j+2}) &= \frac{d}{dx_{j,k}} C_x(t_j, t_{j+2}) = \frac{d}{dx_{j,k}} \left(C_x(t_{j+1}, t_{j+2}) \cdot C_x(t_j, t_{j+1})\right) \\ &= \frac{d}{dx_{j,k}} \left(C_x(t_{j+1}, t_{j+2})\right) \cdot C_x(t_j, t_{j+1}) \\ &\quad + C_x(t_{j+1}, t_{j+2}) \cdot \frac{d}{dx_{j,k}} \left(C_x(t_j, t_{j+1})\right). \end{aligned} \tag{6.60}$$

With the notation $C = \{A \cdot b\}$ for the product of a three-dimensional tensor A with a vector b resulting in a matrix C we can write:

$$D_{x,k}(t_j, t_{j+2}) = \{D_x(t_{j+1}, t_{j+2}) \cdot C_{x,k}(t_j, t_{j+1})\} \cdot C_x(t_j, t_{j+1}) + C_x(t_{j+1}, t_{j+2}) \cdot D_{x,k}(t_j, t_{j+1}). \tag{6.61}$$

For three intervals $[t_j, t_{j+1}]$, $[t_{j+1}, t_{j+2}]$, $[t_{j+2}, t_{j+3}]$ we would equivalently obtain

$$\begin{aligned} D_{x,k}(t_j, t_{j+3}) &= \{D_x(t_{j+2}, t_{j+3}) \cdot (C_x(t_{j+1}, t_{j+2}) \cdot C_{x,k}(t_j, t_{j+1}))\} \\ &\quad \cdot C_x(t_{j+1}, t_{j+2}) \cdot C_x(t_j, t_{j+1}) \\ &\quad + C_x(t_{j+2}, t_{j+3}) \cdot \{D_x(t_{j+1}, t_{j+2}) \cdot C_{x,k}(t_j, t_{j+1})\} \cdot C_x(t_j, t_{j+1}) \\ &\quad + C_x(t_{j+2}, t_{j+3}) \cdot C_x(t_{j+1}, t_{j+2}) \cdot D_{x,k}(t_j, t_{j+1}). \end{aligned} \tag{6.62}$$

Derivative with Respect to Parameters

The k-th component of the derivative of the two-interval monodromy matrix with respect to parameters is

$$\begin{aligned} D_{p,k}(t_j, t_{j+2}) &= \frac{d}{dp_k} C_x(t_j, t_{j+2}) = \frac{d}{dp_k} \left(C_x(t_{j+1}, t_{j+2}) \cdot C_x(t_j, t_{j+1})\right) \\ &= \frac{d}{dp_k} \left(C_x(t_{j+1}, t_{j+2})\right) \cdot C_x(t_j, t_{j+1}) + C_x(t_{j+1}, t_{j+2}) \cdot \frac{d}{dp_k} \left(C_x(t_j, t_{j+1})\right) \\ &= \left(D_{p,k}(t_{j+1}, t_{j+2}) + \{D_x(t_{j+1}, t_{j+2}) \cdot C_{p,k}(t_j, t_{j+1})\}\right) \cdot C_x(t_j, t_{j+1}) \\ &\quad + C_x(t_{j+1}, t_{j+2}) \cdot D_{p,k}(t_j, t_{j+1}). \end{aligned} \tag{6.63}$$

6.6.2 Discontinuous Dynamics

Now we focus on the more complex case of state-dependent discontinuities in the monodromy matrix. We use the same terminology as in section 6.4 and figure 6.4.

Derivative with Respect to Initial Values

Using equation (6.51) and the notation $x_i = x(t_i)$ we can state for the k-th component of the derivative

$$\begin{aligned}\frac{d}{dx_k(t_j)}\left(C_x(t_j,t_{j+1})\right) = \frac{d}{dx_{j,k}}\left(\frac{dx_{j+1}}{dx_j}\right) = &\underbrace{\frac{d}{dx_{j,k}}\left(C_x(t_s^+,t_{j+1})\right)}_{=:\mathcal{D}}\cdot U_x\cdot C_x(t_j,t_s^-)\\ &+ C_x(t_s^+,t_{j+1})\cdot\underbrace{\frac{d}{dx_{j,k}}\left(U_x\right)}_{=:\mathcal{E}}\cdot C_x(t_j,t_s^-)\\ &+ C_x(t_s^+,t_{j+1})\cdot U_x\cdot\underbrace{\frac{d}{dx_{j,k}}\left(C_x(t_j,t_s^-)\right)}_{=:\mathcal{F}}\end{aligned} \tag{6.64}$$

We can now independently derive expressions for terms $\mathcal{D}$, $\mathcal{E}$, and $\mathcal{F}$. With

$$C_x(t_s^+,t_{j+1}) = C_x(t_{j+1};t_s^+(x_j,p),x(t_s^+),p) \tag{6.65}$$

and the dependency of $x(t_s^+)$ defined by equation (6.43) we obtain for $\mathcal{D}$

$$\begin{aligned}\mathcal{D} &= \frac{d}{dx_{j,k}}\left(C_x(t_s^+,t_{j+1})\right)\\ &= \frac{\partial}{\partial t_s}\left(C_x(t_s^+,t_{j+1})\right)\cdot\frac{dt_s}{dx_{j,k}} + \left\{\frac{\partial}{\partial x_s^+}\left(C_x(t_s^+,t_{j+1})\right)\cdot\frac{dx_s^+}{dx_{j,k}}\right\}.\end{aligned} \tag{6.66}$$

Using equations (6.49) and (6.50) and

$$\frac{\partial}{\partial t_s}\left(C_x(t_s^+,t_{j+1})\right) = \left\{\frac{\partial^2 x_{j+1}}{\partial x_s{+}^2}\cdot\dot{x}(t_s^+)\right\} = \left\{D_x(t_s^+,t_{j+1})\cdot\dot{x}(t_s^+)\right\} \tag{6.67}$$

we have

$$\begin{aligned}\mathcal{D} &= \left\{D_x(t_s^+,t_{j+1})\cdot\dot{x}(t_s^+)\right\}\cdot\frac{-s_x(t_s^-)^T}{\dot{s}(t_s^-)}\cdot C_{x,k}(t_j,t_s^-) + \Big\{D_x(t_s^+,t_{j+1})\cdot\\ &\quad\left(\left(\dot{x}(t_s^-)+J_t+J_x\dot{x}(t_s^-)\right)\cdot\frac{-s_x(t_s^-)^T}{\dot{s}(t_s^-)} + (I+J_x)\right)\cdot C_{x,k}(t_j,t_s^-)\Big\}.\end{aligned} \tag{6.68}$$

For the computation of term $\mathcal{E}$

$$\begin{aligned}\mathcal{E} &= \left(\frac{df(t_s^+)}{dx_{j,k}} - \frac{df(t_s^-)}{dx_{j,k}} - \frac{dJ_x}{dx_{j,k}}f(t_s^-) - J_x\frac{df(t_s^-)}{dx_{j,k}} - \frac{dJ_t}{dx_{j,k}}\right)\frac{s_x^T}{\dot{s}}\\ &\quad + \left(f(t_s^+) - f(t_s^-) - J_x f(t_s^-) - J_t\right)\frac{\left(\frac{ds_x}{dx_{j,k}}\right)^T\cdot\dot{s} - s_x^T\cdot\frac{d\dot{s}}{dx_{j,k}}}{\dot{s}^2} + \frac{dJ_x}{dx_{j,k}}\end{aligned} \tag{6.69}$$

we need the derivatives of $f(t_s^-)$, $f(t_s^+)$, J_x, J_t, s_x, s_t, and $\dot{s}$ with respect to $x_{j,k}$:

$$\frac{df(t_s^-)}{dx_{j,k}} = \frac{d\dot{x}(t_s^-)}{dx_{j,k}} = \Big(f_t(t_s^-) + f_x(t_s^-)\cdot\dot{x}(t_s^-)\Big)\cdot\frac{dt_s}{dx_{j,k}} + f_x(t_s^-)\cdot C_{x,k}(t_j,t_s^-) \quad (6.70)$$

$$\begin{aligned}\frac{df(t_s^+)}{dx_{j,k}} &= f_t(t_s^+)\frac{dt_s}{dx_{j,k}} + f_x(t_s^+)\frac{dx_s^+}{dx_{j,k}}\\ &= \big(f_t(t_s^+) + f_x(t_s^+)\cdot\big(\dot{x}(t_s^-) + J_t + J_x\dot{x}(t_s^-)\big)\big)\cdot\frac{dt_s}{dx_{j,k}}\\ &\quad + f_x(t_s^+)\cdot(I+J_x)\cdot C_{x,k}(t_j,t_s^-)\end{aligned} \quad (6.71)$$

$$\frac{dJ_x}{dx_{j,k}} = \Big(J_{xt} + \big\{J_{xx}\cdot\dot{x}(t_s^-)\big\}\Big)\cdot\frac{dt_s}{dx_{j,k}} + J_{xx}\cdot C_{x,k}(t_j,t_s^-) \quad (6.72)$$

$$\frac{dJ_t}{dx_{j,k}} = \Big(J_{tt} + J_{tx}^T\cdot\dot{x}(t_s^-)\Big)\cdot\frac{dt_s}{dx_{j,k}} + J_{xt}^T\cdot C_{x,k}(t_j,t_s^-) \quad (6.73)$$

$$\frac{ds_x}{dx_{j,k}} = \Big(s_{xt} + s_{xx}\cdot\dot{x}(t_s^-)\Big)\cdot\frac{dt_s}{dx_{j,k}} + s_{xx}\cdot C_{x,k}(t_j,t_s^-) \quad (6.74)$$

$$\frac{ds_t}{dx_{j,k}} = \Big(s_{tt} + s_{tx}^T\cdot\dot{x}(t_s^-)\Big)\cdot\frac{dt_s}{dx_{j,k}} + s_{tx}^T\cdot C_{x,k}(t_j,t_s^-) \quad (6.75)$$

$$\begin{aligned}\frac{d\dot{s}}{dx_{j,k}} &= \frac{ds_t}{dx_{j,k}} + (\frac{ds_x}{dx_{j,k}})^T\cdot\dot{x}(t_s^-) + s_x^T\cdot\frac{d\dot{x}(t_s^-)}{dx_{j,k}}\\ &= \Big(s_{tt} + s_{tx}^T\cdot\dot{x}(t_s^-)\Big)\cdot\frac{dt_s}{dx_{j,k}} + s_{tx}^T\cdot C_{x,k}(t_j,t_s^-) +\\ &\quad \Big(\Big(s_{xt} + s_{xx}\cdot\dot{x}(t_s^-)\Big)\cdot\frac{dt_s}{dx_{j,k}} + s_{xx}\cdot C_{x,k}(t_j,t_s^-)\Big)^T\cdot f(t_s^-)\\ &\quad + s_x^T\Big(\Big(f_t(t_s^-) + f_x(t_s^-)\cdot\dot{x}(t_s^-)\Big)\cdot\frac{dt_s}{dx_{j,k}} + f_x(t_s^-)\cdot C_{x,k}(t_j,t_s^-)\Big)\end{aligned} \quad (6.76)$$

and thus obtain

$$\begin{aligned}\mathcal{E} &= \Big(\big(f_t(t_s^+) + f_x(t_s^+)\cdot\big(f(t_s^-) + J_t + J_x f(t_s^-)\big)\big)\cdot\frac{dt_s}{dx_{j,k}}\\ &\quad + f_x(t_s^+)\cdot(I+J_x)\cdot C_{x,k}(t_j,t_s^-)\\ &\quad -\Big(f_t(t_s^-) + f_x(t_s^-)\cdot f(t_s^-)\Big)\cdot\frac{dt_s}{dx_{j,k}} + f_x(t_s^-)\cdot C_{x,k}(t_j,t_s^-)\\ &\quad -\Big(\Big(J_{xt} + \big\{J_{xx}\cdot f(t_s^-)\big\}\Big)\cdot\frac{dt_s}{dx_{j,k}} + J_{xx}\cdot C_{x,k}(t_j,t_s^-)\Big)\cdot f(t_s^-)\\ &\quad -J_x\Big(\Big(f_t(t_s^-) + f_x(t_s^-)\cdot f(t_s^-)\Big)\cdot\frac{dt_s}{dx_{j,k}} + f_x(t_s^-)\cdot C_{x,k}(t_j,t_s^-)\Big)\\ &\quad -\Big(J_{tt} + J_{tx}^T\cdot f(t_s^-)\Big)\cdot\frac{dt_s}{dx_{j,k}} + J_{xt}^T\cdot C_{x,k}(t_j,t_s^-)\Big)\cdot\frac{s_x^T}{\dot{s}}\\ &\quad +\Big(f(t_s^+) - f(t_s^-) - J_x f(t_s^-) - J_t\Big)\frac{1}{\dot{s}^2}\\ &\quad \Big(\Big(\Big(s_{xt} + s_{xx}\cdot f(t_s^-)\Big)\cdot\frac{dt_s}{dx_{j,k}} + s_{xx}\cdot C_{x,k}(t_j,t_s^-)\Big)^T\cdot\dot{s}\end{aligned}$$

$$
\begin{aligned}
&- s_x^T \cdot \Big(\Big(s_{tt} + s_{tx}^T \cdot f(t_s^-) \Big) \cdot \frac{dt_s}{dx_{j,k}} + s_{tx}^T \cdot C_{x,k}(t_j, t_s^-) + \\
&\Big(\Big(s_{xt} + s_{xx} \cdot f(t_s^-) \Big) \cdot \frac{dt_s}{dx_{j,k}} + s_{xx} \cdot C_{x,k}(t_j, t_s^-) \Big)^T \cdot f(t_s^-) \\
&+ s_x^T \cdot \Big(\Big(f_t(t_s^-) + f_x(t_s^-) \cdot f(t_s^-) \Big) \cdot \frac{dt_s}{dx_{j,k}} + f_x(t_s^-) \cdot C_{x,k}(t_j, t_s^-) \Big) \Big) \Big) \\
&+ \Big(J_{xt} + \big\{ J_{xx} \cdot f(t_s^-) \big\} \Big) \cdot \frac{dt_s}{dx_{j,k}} + \big\{ J_{xx} \cdot C_{x,k}(t_j, t_s^-) \big\} . \qquad (6.77)
\end{aligned}
$$

$\mathcal{F}$ is simply

$$\mathcal{F} = D_{x,k}(t_j, t_s^-). \qquad (6.78)$$

Overall, this leads to the following expression for the second derivative of the dynamics with respect to initial values in the presence of discontinuities:

$$
\begin{aligned}
&\frac{d}{dx_k(t_j)} \left(C_x(t_j, t_{j+1}) \right) = \\
&\Bigg(\big\{ D_x(t_s^+, t_{j+1}) \cdot f(t_s^+) \big\} \cdot \frac{-s_x(t_s^-)^T}{\dot{s}(t_s^-)} \cdot C_{x,k}(t_j, t_s^-) + \Big\{ D_x(t_s^+, t_{j+1}) \cdot \\
&\Big(\big(f(t_s^-) + J_t + J_x f(t_s^-) \big) \cdot \frac{-s_x(t_s^-)^T}{\dot{s}(t_s^-)} + (I + J_x) \Big) \cdot C_{x,k}(t_j, t_s^-) \Big\} \Bigg) \cdot U_x \cdot C_x(t_j, t_s^+) \\
&+ C_x(t_s^+, t_{j+1}) \cdot \Bigg(\Bigg(\big(f_t(t_s^+) + f_x(t_s^+) \cdot \big(f(t_s^-) + J_t + J_x f(t_s^-) \big) \big) \cdot \frac{dt_s}{dx_{j,k}} \\
&+ f_x(t_s^+) \cdot (I + J_x) \cdot C_{x,k}(t_j, t_s^-) - \Big(f_t(t_s^-) + f_x(t_s^-) \cdot f(t_s^-) \Big) \cdot \frac{dt_s}{dx_{j,k}} \\
&+ f_x(t_s^-) \cdot C_{x,k}(t_j, t_s^-) - \Big(\Big(J_{xt} + \big\{ J_{xx} \cdot f(t_s^-) \big\} \Big) \cdot \frac{dt_s}{dx_{j,k}} + J_{xx} \cdot C_{x,k}(t_j, t_s^-) \Big) \cdot f(t_s^-) \\
&- J_x \Big(\Big(f_t(t_s^-) + f_x(t_s^-) \cdot f(t_s^-) \Big) \cdot \frac{dt_s}{dx_{j,k}} + f_x(t_s^-) \cdot C_{x,k}(t_j, t_s^-) \Big) \\
&- \Big(J_{tt} + J_{tx}^T \cdot f(t_s^-) \Big) \cdot \frac{dt_s}{dx_{j,k}} + J_{xt}^T \cdot C_{x,k}(t_j, t_s^-) \Big) \cdot \frac{s_x^T}{\dot{s}} \\
&+ \Big(f(t_s^+) - f(t_s^-) - J_x f(t_s^-) - J_t \Big) \frac{1}{\dot{s}^2} \\
&\Big(\Big(\Big(s_{xt} + s_{xx} \cdot f(t_s^-) \Big) \cdot \frac{dt_s}{dx_{j,k}} + s_{xx} \cdot C_{x,k}(t_j, t_s^-) \Big)^T \cdot \dot{s} \\
&- s_x^T \cdot \Big(\Big(s_{tt} + s_{tx}^T \cdot f(t_s^-) \Big) \cdot \frac{dt_s}{dx_{j,k}} + s_{tx}^T \cdot C_{x,k}(t_j, t_s^-) + \\
&\Big(\Big(s_{xt} + s_{xx} \cdot f(t_s^-) \Big) \cdot \frac{dt_s}{dx_{j,k}} + s_{xx} \cdot C_{x,k}(t_j, t_s^-) \Big)^T \cdot f(t_s^-) \\
&+ s_x^T \cdot \Big(\Big(f_t(t_s^-) + f_x(t_s^-) \cdot f(t_s^-) \Big) \cdot \frac{dt_s}{dx_{j,k}} + f_x(t_s^-) \cdot C_{x,k}(t_j, t_s^-) \Big) \Big) \Big) \\
&+ \Big(J_{xt} + \big\{ J_{xx} \cdot f(t_s^-) \big\} \Big) \cdot \frac{dt_s}{dx_{j,k}} + \big\{ J_{xx} \cdot C_{x,k}(t_j, t_s^-) \big\} \Big) \cdot C_x(t_j, t_s^+)
\end{aligned}
$$

$$+ C_x(t_s^+, t_{j+1}) \cdot U_x \cdot D_{x,k}(t_j, t_s^+). \tag{6.79}$$

This formula has not been implemented yet. In view of its complexity it is very questionable if this approach will be relevant for the solution of practical problems or if e.g. external finite difference schemes should be favored for second order derivatives. The same is true for the derivative with respect to parameters (6.97) derived in the following section.

Derivative with Respect to Parameters

The derivative of the monodromy matrix with respect to parameters is derivable by a similar procedure. Its k-th component is

$$\begin{aligned} \frac{d}{dp_k}\left(C_x(t_j, t_{j+1})\right) = \frac{d}{dp_k}\left(\frac{dx_{j+1}}{dx_j}\right) &= \underbrace{\frac{d}{dp_k}\left(C_x(t_s^+, t_{j+1})\right)}_{=:\mathcal{G}} \cdot U_x \cdot C_x(t_j, t_s^-) \\ &\quad + C_x(t_s^+, t_{j+1}) \cdot \underbrace{\frac{d}{dp_k}\left(U_x\right)}_{=:\mathcal{H}} \cdot C_x(t_j, t_s^-) \\ &\quad + C_x(t_s^+, t_{j+1}) \cdot U_x \cdot \underbrace{\frac{d}{dp_k}\left(C_x(t_j, t_s^-)\right)}_{=:\mathcal{I}} \end{aligned} \tag{6.80}$$

with the unknown terms $\mathcal{G}$, $\mathcal{H}$, and $\mathcal{I}$.

The first term, $\mathcal{G}$, is

$$\mathcal{G} = \frac{\partial}{\partial t_s}\left(C_x(t_s^+, t_{j+1})\right) \cdot \frac{dt_s}{dp_k} + \left\{\frac{\partial}{\partial x_s^+}\left(C_x(t_s^+, t_{j+1})\right) \cdot \frac{dx_s^+}{dp_k}\right\} + \frac{\partial}{\partial p_k}\left(C_x(t_s^+, t_{j+1})\right). \tag{6.81}$$

Thus, we must first compute $\frac{dt_s}{dp_k}$ and $\frac{dx_s^+}{dp_k}$.

In analogy to (6.47) – (6.49) we can derive

$$\frac{ds}{dp} = s_t(t_s^-)\frac{dt_s}{dp} + s_x(t_s^-)\dot{x}(t_s^-)\frac{dt_s}{dp} + s_x(t_s^-) \cdot \frac{dx(t_s^-)}{dx_j} + s_p(t_s^-) = 0 \tag{6.82}$$

$$\underbrace{\left(s_t(t_s^-) + s_x(t_s^-)\dot{x}(t_s^-)\right)}_{=:\dot{s}(t_s^-)}\frac{dt_s}{dp} + s_x(t_s^-) \cdot C_p(t_j, t_s^-) + s_p(t_s^-) = 0 \tag{6.83}$$

$$\frac{dt_s}{dp} = \frac{-1}{\dot{s}(t_s^-)} \cdot \left(s_x(t_s^-) \cdot C_p(t_j, t_s^-) + s_p(t_s^-)\right). \tag{6.84}$$

Using equation (6.50), we have

$$\frac{dx_s^+}{dp_k} = \dot{x}(t_s^-)\frac{dt_s}{dp_k} + \frac{\partial x(t_s^-)}{\partial p_k} + J_t\frac{dt_s}{dp_k} + J_x\left(\dot{x}(t_s^-)\frac{dt_s}{dp_k} + \frac{\partial x(t_s^-)}{\partial p_k}\right) + J_{p,k}$$

$$
\begin{aligned}
&= \big(\dot{x}(t_s^-) + J_t + J_x\dot{x}(t_s^-)\big)\cdot\frac{-1}{\dot{s}(t_s^-)}\cdot\big(s_x(t_s^-)\cdot C_{p,k}(t_j,t_s^-) + s_{p,k}(t_s^-)\big)\\
&\quad + (I+J_x)\cdot C_{p,k}(t_j,t_s^-) + J_{p,k}. \qquad (6.85)
\end{aligned}
$$

With this we find

$$
\begin{aligned}
\mathcal{G} &= \big\{D_x(t_s^+,t_{j+1})\cdot\dot{x}(t_s^+)\big\}\cdot\frac{-1}{\dot{s}(t_s^-)}\cdot\big(s_x(t_s^-)\cdot C_p(t_j,t_s^-) + s_p(t_s^-)\big)\\
&\quad + \Big\{D_{x,k}(t_s^+,t_{j+1})\cdot\Big(\big(\dot{x}(t_s^-) + J_t + J_x\dot{x}(t_s^-)\big)\cdot\frac{-1}{\dot{s}(t_s^-)}\cdot\big(s_x(t_s^-)\cdot C_{p,k}(t_j,t_s^-) + s_{p,k}(t_s^-)\big)\\
&\quad + (I+J_x)\cdot C_{p,k}(t_j,t_s^-) + J_{p,k}\Big)\Big\} + D_{p.k}(t_s^+,t_{j+1}) \qquad (6.86)
\end{aligned}
$$

For the second term, H

$$
\begin{aligned}
\mathcal{H} &= \left(\frac{df(t_s^+)}{dp_k} - \frac{df(t_s^-)}{dp_k} - \frac{dJ_x}{dp_k}\dot{x}(t_s^-) - J_x\frac{df(t_s^-)}{dp_k} - \frac{dJ_t}{dp_k}\right)\frac{s_x^T}{\dot{s}}\\
&\quad + \big(f(t_s^+) - f(t_s^-) - J_x f(t_s^-) - J_t\big)\frac{(\frac{ds_x}{dp_k})^T\cdot\dot{s} - s_x^T\cdot\frac{d\dot{s}}{dp_k}}{\dot{s}^2} + \frac{dJ_x}{dp_k} \qquad (6.87)
\end{aligned}
$$

we need the derivatives of $f(t_s^-)$, $f(t_s^+)$, J_x, J_t, s_x, s_t, and $\dot{s}$ with respect to p_k:

$$
\begin{aligned}
\frac{df(t_s^-)}{dp_k} &= \frac{d\dot{x}(t_s^-)}{dp_k} = \Big(f_t(t_s^-) + f_x(t_s^-)\dot{x}(t_s^-)\Big)\cdot\frac{dt_s}{dp_k}\\
&\quad + f_x(t_s^-)\cdot C_{p,k}(t_j,t_s^-) + f_{p,k}(t_s^-) \qquad (6.88)\\
\frac{df(t_s^+)}{dp_k} &= f_t(t_s^+)\frac{dt_s}{dp_k} + f_x(t_s^+)\Big(\dot{x}(t_s^-)\frac{dt_s}{dp_k} + C_{p,k}(t_j,t_s^-) + J_t\frac{dt_s}{dp_k}\\
&\quad + J_x\Big(\dot{x}(t_s^-)\frac{dt_s}{dp_k} + C_{p,k}(t_j,t_s^-)\Big) + J_{p,k}\Big) + f_{p,k}(t_s^+)\\
&= \Big(f_t(t_s^+) + f_x(t_s^+)\Big(\dot{x}(t_s^-) + J_t + J_x\dot{x}(t_s^-)\Big)\Big)\cdot\frac{dt_s}{dp_k}\\
&\quad + f_x(t_s^-)\Big(I + J_x\Big)C_{p,k}(t_j,t_s^-) + f_{p,k}(t_s^+) \qquad (6.89)\\
\frac{dJ_x}{dp_k} &= \Big(J_{xt} + \big\{J_{xx}\cdot\dot{x}(t_s^-)\big\}\Big)\cdot\frac{dt_s}{dp_k} + \big\{J_{xx}\cdot C_{p,k}(t_j,t_s^-)\big\} + J_{x\,p,k} \qquad (6.90)\\
\frac{dJ_t}{dp_k} &= \Big(J_{tt} + J_{tx}^T\cdot\dot{x}(t_s^-)\Big)\cdot\frac{dt_s}{dp_k} + J_{xt}^T\cdot C_{p,k}(t_j,t_s^-) + J_{t\,p,k} \qquad (6.91)\\
\frac{ds_x}{dp_k} &= \Big(s_{xt} + s_{xx}\cdot\dot{x}(t_s^-)\Big)\cdot\frac{dt_s}{dp_k} + s_{xx}\cdot C_{p,k}(t_j,t_s^-) + s_{x\,p,k} \qquad (6.92)\\
\frac{ds_t}{dp_k} &= \Big(s_{tt} + s_{tx}^T\cdot\dot{x}(t_s^-)\Big)\cdot\frac{dt_s}{dp_k} + s_{tx}^T\cdot C_{p,k}(t_j,t_s^-) + s_{t\,p,k} \qquad (6.93)\\
\frac{d\dot{s}}{dp_k} &= \frac{ds_t}{dp_k} + (\frac{ds_x}{dp_k})^T\cdot\dot{x}(t_s^-) + s_x^T\cdot\frac{d\dot{x}(t_s^-)}{dp_k}\\
&= \Big(s_{tt} + s_{tx}^T\cdot\dot{x}(t_s^-)\Big)\cdot\frac{dt_s}{dp_k} + s_{tx}^T\cdot C_{p,k}(t_j,t_s^-) + s_{t\,p,k} +
\end{aligned}
$$

$$\Big(\Big(s_{xt} + s_{xx} \cdot \dot{x}(t_s^-)\Big) \cdot \frac{dt_s}{dp_k} + s_{xx} \cdot C_{p,k}(t_j, t_s^-) + s_{x\,p,k}\Big)^T \cdot f(t_s^-)$$
$$+ s_x^T \Big(\Big(f_t(t_s^-) + f_x(t_s^-)\dot{x}(t_s^-)\Big)\frac{dt_s}{dp_k} + \; f_x(t_s^-) \cdot C_{p,k}(t_j, t_s^-) + f_p(t_s^-)\Big) \quad (6.94)$$

$$\begin{aligned}
\mathcal{H} \;=\; & \Big(\Big(f_t(t_s^+) + f_x(t_s^+)\Big(\dot{x}(t_s^-) + J_t + J_x\dot{x}(t_s^-)\Big)\Big) \cdot \frac{-1}{\dot{s}(t_s^-)} \cdot \big(s_x(t_s^-) \cdot C_{p,k}(t_j, t_s^-) + s_{p,k}(t_s^-)\big) \\
& + f_x(t_s^-)\Big(I + J_x\Big)C_{p,k}(t_j, t_s^-) + f_{p,k}(t_s^+) \\
& - \Big(f_t(t_s^-) + f_x(t_s^-)\dot{x}(t_s^-)\Big) \cdot \frac{-1}{\dot{s}} \cdot \big(s_x \cdot C_{p,k}(t_j, t_s^-) + s_{p,k}\big) \\
& + f_x(t_s^-) \cdot C_{p,k}(t_j, t_s^-) + f_{p,k}(t_s^-) \\
& - \Big(\Big(J_{xt} + \big\{J_{xx} \cdot \dot{x}(t_s^-)\big\}\Big) \cdot \frac{-1}{\dot{s}} \cdot \big(s_x \cdot C_{p,k}(t_j, t_s^-) + s_{p,k}\big) \\
& + \big\{J_{xx} \cdot C_{p,k}(t_j, t_s^-)\big\} + J_{x\,p,k}\Big) \cdot f(t_s^-) \\
& - J_x\Big(\Big(f_t(t_s^-) + f_x(t_s^-)\dot{x}(t_s^-)\Big) \cdot \frac{-1}{\dot{s}} \cdot \big(s_x \cdot C_{p,k}(t_j, t_s^-) + s_{p,k}\big) \\
& + f_x(t_s^-) \cdot C_{p,k}(t_j, t_s^-) + f_{p,k}(t_s^-)\Big) - \Big(J_{tt} + J_{tx}^T \cdot \dot{x}(t_s^-)\Big) \cdot \frac{-1}{\dot{s}} \cdot \big(s_x \cdot C_{p,k}(t_j, t_s^-) + s_{p,k}\big) \\
& + J_{xt}^T \cdot C_{p,k}(t_j, t_s^-) + J_{t\,p,k}\Big)\frac{s_x^T}{\dot{s}} + \; \Big(f(t_s^+) - f(t_s^-) - J_x f(t_s^-) - J_t\Big)\frac{1}{\dot{s}^2} \\
& \cdot \Big(\Big(\Big(s_{xt} + s_{xx} \cdot \dot{x}(t_s^-)\Big) \cdot \frac{-1}{\dot{s}} \cdot \big(s_x \cdot C_{p,k}(t_j, t_s^-) + s_{p,k}\big) + s_{xx} \cdot C_{p,k}(t_j, t_s^-) + s_{x\,p,k}\Big)^T \cdot \dot{s} \\
& - s_x^T \cdot \Big(\Big(s_{tt} + s_{tx}^T \cdot \dot{x}(t_s^-)\Big) \cdot \frac{-1}{\dot{s}} \cdot \big(s_x \cdot C_{p,k}(t_j, t_s^-) + s_{p,k}\big) + s_{tx}^T \cdot C_{p,k}(t_j, t_s^-) + s_{t\,p,k} + \\
& \Big(\Big(s_{xt} + s_{xx} \cdot \dot{x}(t_s^-)\Big) \cdot \frac{dt_s}{dp_k} + s_{xx} \cdot C_{p,k}(t_j, t_s^-) + s_{x\,p,k}\Big)^T \cdot f(t_s^-) \\
& + s_x^T \cdot \Big(\Big(f_t(t_s^-) + f_x(t_s^-)\dot{x}(t_s^-)\Big) \cdot \frac{-1}{\dot{s}} \cdot \big(s_x \cdot C_{p,k}(t_j, t_s^-) + s_{p,k}\big) \\
& + f_x(t_s^-) \cdot C_{p,k}(t_j, t_s^-) + f_p(t_s^-)\Big)\Big)\Big) + \Big(J_{xt} + \big\{J_{xx} \cdot \dot{x}(t_s^-)\big\}\Big) \cdot \\
& \frac{-1}{\dot{s}} \cdot \big(s_x \cdot C_{p,k}(t_j, t_s^-) + s_{p,k}\big) + \big\{J_{xx} \cdot C_{p,k}(t_j, t_s^-)\big\} + J_{x\,p,k} \qquad (6.95)
\end{aligned}$$

The third term is again the most simple one:

$$\mathcal{I} = D_{p,k}(t_j, t_s^-). \quad (6.96)$$

Collecting all this, we obtain for the derivative of the monodromy matrix with respect to parameters in the presence of discontinuities:

$$\frac{d}{dp_k}\left(C_x(t_j, t_{j+1})\right) =$$

$$
\begin{aligned}
&\Big(\big\{D_x(t_s^+, t_{j+1}) \cdot \dot{x}(t_s^+)\big\} \cdot \frac{-1}{\dot{s}(t_s^-)} \cdot \big(s_x(t_s^-) \cdot C_p(t_j, t_s^-) + s_p(t_s^-)\big) \\
&+ \Big\{ D_{x,k}(t_s^+, t_{j+1}) \cdot \Big(\big(\dot{x}(t_s^-) + J_t + J_x \dot{x}(t_s^-)\big) \cdot \frac{-1}{\dot{s}(t_s^-)} \cdot \big(s_x(t_s^-) \cdot C_{p,k}(t_j, t_s^-) + s_{p,k}(t_s^-)\big) \\
&\quad + (I + J_x) \cdot C_{p,k}(t_j, t_s^-) + J_{p,k} \Big) \Big\} + D_{p.k}(t_s^+, t_{j+1}) \Big) \cdot U_x \cdot C_x(t_j, t_s^-) \\
&+ C_x(t_s^+, t_{j+1}) \cdot \Big(\Big(\Big(f_t(t_s^+) + f_x(t_s^+) \Big(\dot{x}(t_s^-) + J_t + J_x \dot{x}(t_s^-)\Big)\Big) \cdot \\
&\frac{-1}{\dot{s}(t_s^-)} \cdot \big(s_x(t_s^-) \cdot C_{p,k}(t_j, t_s^-) + s_{p,k}(t_s^-)\big) \\
&+ f_x(t_s^-)\Big(I + J_x\Big) C_{p,k}(t_j, t_s^-) + f_{p,k}(t_s^+) \\
&- \Big(f_t(t_s^-) + f_x(t_s^-) \dot{x}(t_s^-)\Big) \cdot \frac{-1}{\dot{s}} \cdot \big(s_x \cdot C_{p,k}(t_j, t_s^-) + s_{p,k}\big) \\
&+ f_x(t_s^-) \cdot C_{p,k}(t_j, t_s^-) + f_{p,k}(t_s^-) \\
&- \Big(\Big(J_{xt} + \big\{J_{xx} \cdot \dot{x}(t_s^-)\big\} \Big) \cdot \frac{-1}{\dot{s}} \cdot \big(s_x \cdot C_{p,k}(t_j, t_s^-) + s_{p,k}\big) \\
&+ \big\{J_{xx} \cdot C_{p,k}(t_j, t_s^-)\big\} + J_{x\,p,k} \Big) \cdot f(t_s^-) \\
&- J_x \Big(\Big(f_t(t_s^-) + f_x(t_s^-) \dot{x}(t_s^-)\Big) \cdot \frac{-1}{\dot{s}} \cdot \big(s_x \cdot C_{p,k}(t_j, t_s^-) + s_{p,k}\big) \\
&+ f_x(t_s^-) \cdot C_{p,k}(t_j, t_s^-) + f_{p,k}(t_s^-)\Big) - \Big(J_{tt} + J_{tx}^T \cdot \dot{x}(t_s^-)\Big) \cdot \frac{-1}{\dot{s}} \cdot \big(s_x \cdot C_{p,k}(t_j, t_s^-) + s_{p,k}\big) \\
&+ J_{xt}^T \cdot C_{p,k}(t_j, t_s^-) + J_{t\,p,k} \Big) \frac{s_x^T}{\dot{s}} + \Big(f(t_s^+) - f(t_s^-) - J_x f(t_s^-) - J_t \Big) \frac{1}{\dot{s}^2} \\
&\cdot \Big(\Big(\Big(s_{xt} + s_{xx} \cdot \dot{x}(t_s^-)\Big) \cdot \frac{-1}{\dot{s}} \cdot \big(s_x \cdot C_{p,k}(t_j, t_s^-) + s_{p,k}\big) + s_{xx} \cdot C_{p,k}(t_j, t_s^-) + s_{x\,p,k} \Big)^T \cdot \dot{s} \\
&- s_x^T \cdot \Big(\Big(s_{tt} + s_{tx}^T \cdot \dot{x}(t_s^-)\Big) \cdot \frac{-1}{\dot{s}} \cdot \big(s_x \cdot C_{p,k}(t_j, t_s^-) + s_{p,k}\big) + s_{tx}^T \cdot C_{p,k}(t_j, t_s^-) + s_{t\,p,k} + \\
&\Big(\Big(s_{xt} + s_{xx} \cdot \dot{x}(t_s^-)\Big) \cdot \frac{dt_s}{dp_k} + s_{xx} \cdot C_{p,k}(t_j, t_s^-) + s_{x\,p,k} \Big)^T \cdot f(t_s^-) \\
&+ s_x^T \cdot \Big(\Big(f_t(t_s^-) + f_x(t_s^-) \dot{x}(t_s^-)\Big) \cdot \frac{-1}{\dot{s}} \cdot \big(s_x \cdot C_{p,k}(t_j, t_s^-) + s_{p,k}\big) \\
&+ f_x(t_s^-) \cdot C_{p,k}(t_j, t_s^-) + f_p(t_s^-) \Big)\Big)\Big) + \Big(J_{xt} + \big\{J_{xx} \cdot \dot{x}(t_s^-)\big\} \Big) \cdot \\
&\frac{-1}{\dot{s}} \cdot \big(s_x \cdot C_{p,k}(t_j, t_s^-) + s_{p,k}\big) + \big\{J_{xx} \cdot C_{p,k}(t_j, t_s^-)\big\} + J_{x\,p,k} \Big) \cdot C_x(t_j, t_s^-) \\
&+ C_x(t_s^+, t_{j+1}) \cdot U_x \cdot D_{p,k}(t_j, t_s^-). \qquad (6.97)
\end{aligned}
$$

6.7 Numerical Determination of Stability Margins

Having found a stable solution by stability optimization based on linear theory it is interesting to ask how well this result captures the stability behavior of the nonlinear system. What is the range of perturbations that can be applied to the above stable solution? There is no theory answering this question but we introduce a numerical criterion.

We study which set of perturbations of initial values as well as of model parameters can be coped with by the robot model and its prescribed control pattern by applying some perturbation and simulating the resulting behavior of the system. The full nonlinear dynamics of the robot are integrated checking if it recovers and persists in its gait or if it stumbles. Two selections have to be made:

- Type of perturbations:
  a) Perturbing the initial values of state variables answers the question how small the 'small perturbations' have to be that linear theory is talking about. We generally apply one-dimensional perturbations to the initial values of all positions and velocities. But for those positions describing an initial phase-separating manifold, like a heelstrike manifold, it is often customary to apply coupled perturbations that are consistent with this manifold.
  b) Even though linear stability theory does not talk about sensitivity with respect to parameters it is interesting to know in what range of parameters gait is possible under the influence of the prescribed controls. We perturb one parameter value at a time keeping the others fixed.
- Integration Interval:
  The choice of the integration interval is quite arbitrary. It has to be long enough to allow unrecoverable perturbations to show their effect and not too long since numerical instabilities would predominate. We have chosen an integration interval of ten physical steps of the robot. Although this choice directly influences the exact result it does not not change the order of magnitude of the stability margins.

For this purpose, an integrator capable to handle switching functions has to be used, e.g. the powerful library ODESIM (see Winckler [102], von Schwerin & Winckler [94]).

Chapter 7

Open-loop Stable One-legged Hopping Robot

The first robot we present in this thesis is a one-legged hopper moving in the vertical plane. The remarkable feature of this robot in contrast to many of its real world relatives (see section 1.1.2) is that despite its flight phase, it neither needs sensors nor sophisticated controllers for stabilization. With its single leg, a small foot and a relatively high center of mass, it has no statically stable standing position.

The robot consists of a toroidal trunk and a telescopic leg coupled by an actuated hinge. The two parts of the leg are connected by an actuated spring-damper element. The foot is fixed to the lower leg without articulation. We have studied circular as well as point shaped feet. The robot can perform stable two-dimensional hopping motions including a non-sliding or rolling contact phase and a flight phase without any feedback controllers. Figure 7.2 shows an animation of one cycle of motion of the hopping robot.

The equivalent "real" robot matching this model has not been built yet. Our model is an extension of the hopping robot of Ringrose [77] presented in section 1.2 to which we have added the trunk with corresponding actuation that makes a periodic forward motion possible. We also have studied the 'hopping in place' motions of the original Ringrose robot and refer to Mombaur et al. [65] for results. This simpler robot is not discussed in this thesis.

We use the example of the hopping robot to study the influence of different stability optimization criteria like the spectral radius and matrix norms. This is the first time solutions for a one-legged hopping robot with point foot are presented. For the model version with circular feet we were able to further improve the stable solution given in Mombaur et al. [63].

Following the classification of section 1.4, the hopping robot is holonomic, but non-conservative due to damper forces and inelastic impacts. The latter property may promote stability of the system.

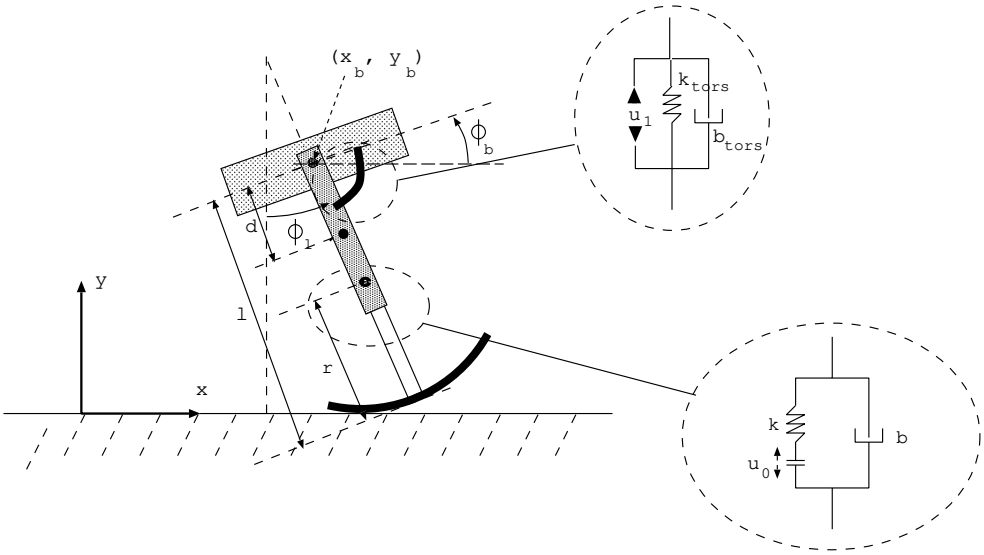


Figure 7.1: Parameters and configuration variables of the hopping robot

7.1 Robot Model

A sketch of the model and its parameters is given in figure 7.1. Parameters are trunk mass and inertia m_b and Θ_b, leg mass and inertia m_l and Θ_l, distance between centers of mass of trunk and leg d, leg rest length l_0, foot radius r, torsional spring and damper constants k_{tors} and b_{tors}, rest location of torsional spring $\Delta\phi$, and translational spring and damper constants k and b. The foot is assumed to be massless. The point foot version is just a special case of the circular foot with $r = 0$.

During the flight phase, the robot has four degrees of freedom. As state variables we choose the uniform set of coordinates

$$q = (x_b, y_b, \phi_b, \phi_l)^T,$$

and the corresponding velocities, where x_b and y_b are two-dimensional position coordinates of the trunk center of mass, and ϕ_b and ϕ_l are the orientations of trunk and leg.

The coordinates of the leg center of mass x_l and y_l can be eliminated using the distance parameter d by

$$x_l = x_b + d \sin \phi_l \tag{7.1}$$
$$y_l = y_b - d \cos \phi_l. \tag{7.2}$$

The leg length l is fixed to $l_0 + u_0$ during the major part of the flight phase (as the foot is massless) and depends on the other coordinates during the contact phase as follows:

$$l = \frac{y_b - r}{\cos \phi_l} + r \tag{7.3}$$

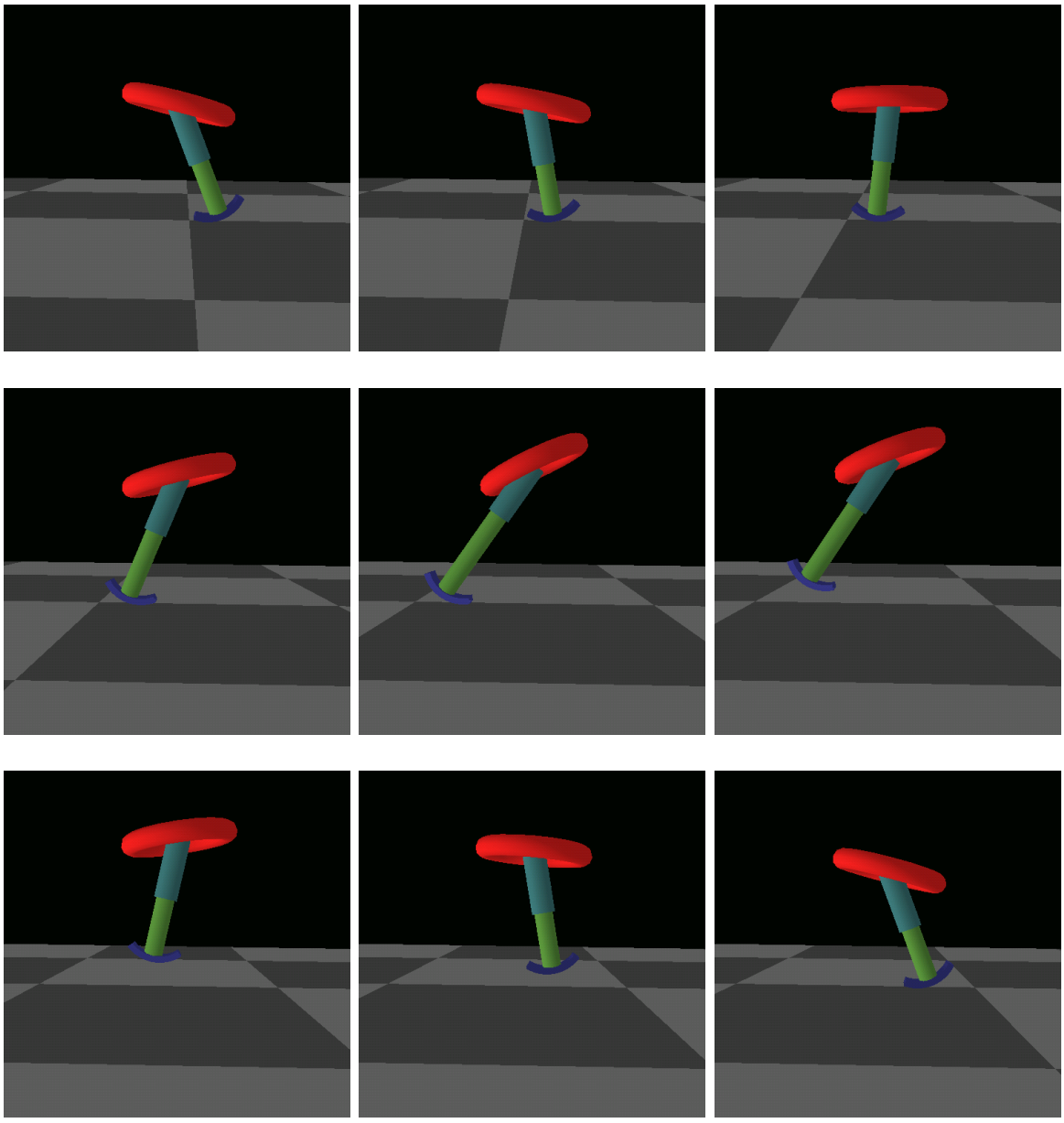


Figure 7.2: Periodic motion of one-legged hopping robot animated with JAFV (Winckler [102], Winckler & Huber [103])

$$\rightarrow \dot{l} \;=\; \frac{\dot{y}_b}{\cos\phi_l} + (y_b - r)\frac{\sin\phi_l}{\cos^2\phi_l}\dot{\phi}_l. \tag{7.4}$$

The robot has two actuators:

1. u_0 – series elastic actuator (SEA) in the prismatic joint :
   as described by Pratt et al. [73], this is an actuated spring-damper element with spring constant k and damping constant b (see figure 7.1). The control $u_0 \geq 0$ actively changes the spring's length which has the same effect as changing the spring's rest length in the opposite direction:

   $$\Delta l = \left(\frac{y_b - r}{\cos\phi_l} + r - u_0\right) - l_0 = \left(\frac{y_b - r}{\cos\phi_l} + r\right) - (l_0 + u_0) \tag{7.5}$$

   The control u_0 is only effective during the contact phase - due to the massless foot it can be brought back to zero position during flight without any effect. u_0 is 0 at touchdown and has to be > 0 at liftoff to compensate for the energy loss in the damper. Instantaneous compressions and general control histories can be modeled.

2. u_1 – torque control between trunk and leg (in parallel with a spring-damper-element k_{tors}, b_{tors}, see figure 7.1).

The one-legged hopping robot is the only robot treated in this thesis for which it is reasonable to give the equations of motion in explicit form. They have been derived using free-body diagrams with all auxiliary coordinates being eliminated.

The equations of motion during the flight phase are described by the following set of ODEs:

$$\begin{pmatrix} m & 0 & 0 & m_l d\cos\phi_l \\ 0 & m & 0 & m_l d\sin\phi_l \\ 0 & 0 & \theta_b & 0 \\ -m_b d\cos\phi_l & -m_b d\sin\phi_l & & \theta_l \end{pmatrix} \begin{pmatrix} \ddot{x}_b \\ \ddot{y}_b \\ \ddot{\phi}_b \\ \ddot{\phi}_l \end{pmatrix} \;=\; \begin{pmatrix} m_l d\sin\phi_l\dot{\phi}_l^2 \\ -m_l d\cos\phi_l\dot{\phi}_l^2 - mg \\ u_1 - k_{tors}(\phi_b - \phi_l - \Delta\phi) - b_{tors}(\dot{\phi}_b - \dot{\phi}_l) + m_b g d\sin\phi_l \\ -u_1 + m_b g d\sin\phi_l + k_{tors}(\phi_b - \phi_l - \Delta\phi) + b_{tors}(\dot{\phi}_b - \dot{\phi}_l) \end{pmatrix} \tag{7.6}$$

where m is the total mass $m = m_b + m_l$ and u_1 is the torque between trunk and leg.

During contact phase we have a superposition of the rolling motion due to the circular foot and the leg length variation influenced by the SEA spring-damper forces. This leads to a reduction from four to three DOFs during contact phase. The coupling is described by the additional kinematic constraint in velocity space

$$\dot{x}_b + (y_b + (y_b - r)\tan^2\phi_l\;)\;\dot{\phi}_l + \tan\phi_l\;\dot{y}_b = 0. \tag{7.7}$$

A corresponding equation for the differences in position space can be formulated. The equations of motion for the contact phase become

$$\begin{pmatrix} m & 0 & 0 & m_l d\cos\phi_l & 1 \\ 0 & m & 0 & m_l d\sin\phi_l & \tan\phi_l \\ 0 & 0 & \theta_b & 0 & 0 \\ -m_b d\cos\phi_l & -m_b d\sin\phi_l & & \theta_l & y_b+(y_b-r)\tan^2\phi_l \\ 1 & \tan\phi_l & 0 & y_b+(y_b-r)\tan^2\phi_l & 0 \end{pmatrix} \begin{pmatrix} \ddot{x}_b \\ \ddot{y}_b \\ \ddot{\phi}_b \\ \ddot{\phi}_l \\ \lambda \end{pmatrix} = \begin{pmatrix} (F_k+F_d)\sin\phi_l + m_l d\sin\phi_l\dot{\phi}_l^2 \\ -(F_k+F_d)\cos\phi_l - m_l d\cos\phi_l\dot{\phi}_l^2 - mg \\ u_1 - k_{tors}(\phi_b-\phi_l-\Delta\phi) - b_{tors}(\dot{\phi}_b-\dot{\phi}_l) + m_b g d\sin\phi_l \\ -u_1 + m_b g d\sin\phi_l + (F_k+F_d)r\sin\phi_l + k_{tors}(\phi_b-\phi_l-\Delta\phi) + b_{tors}(\dot{\phi}_b-\dot{\phi}_l) \\ -2\cdot\cos^{-2}\phi_l\dot{\phi}_l(\dot{y}_b+(y_b-r)\tan\phi_l\dot{\phi}_l \end{pmatrix} \tag{7.8}$$

with spring and damper forces F_k and F_d

$$F_k = k\,(\frac{y_b-r}{\cos\phi_l} + r - l_0 - u_0) \tag{7.9}$$

$$F_d = b\,(\frac{\dot{y}_b}{\cos\phi_l} + (y_b-r)\frac{\tan\phi_l}{\cos\phi_l}\dot{\phi}_l) \tag{7.10}$$

with u_0 being the SEA control. The system of equations (7.8) is a DAE of index 1 derived from an index 3 system by index reduction.

Phase change from contact phase to flight phase (liftoff) takes place, when the spring length is equal to the (modified) rest length:

$$s_{liftoff} = l_0 + u_0 - \frac{y_b-r}{\cos\phi_l} - r = 0 \tag{7.11}$$

and, at the same time, the trunk has a positive vertical speed:

$$c_{liftoff} = \dot{y}_b > 0. \tag{7.12}$$

Touchdown phase change occurs when the height of the prospective contact point is equal to zero

$$s_{touchdown} = y_b - (l_0-r)\cos\phi_l - r = 0. \tag{7.13}$$

The vertical speed of the contact point at touchdown must be negative:

$$c_{touchdown} = \dot{y}_b + (l_0-r)\sin\phi_l\dot{\phi}_l < 0. \tag{7.14}$$

There may be a discontinuity in the velocities at touchdown because friction is assumed to be large enough to instantaneously set the velocity of the contact point equal to zero. There are no jumps in the positions. The four velocities after the touchdown-discontinuity are determined by the following four conditions:

- superposition of rolling motion and spring-damper action:

$$\dot{x}_{contact} = \dot{x}_b + (l_0 - r)\cos\phi_l\dot{\phi}_l + \dot{y}_b\tan\phi_l + (y_b - r)\dot{\phi}_l\tan^2\phi_l = 0 \tag{7.15}$$

- conservation of angular momentum of trunk about contact point:

$$H_{trunk,hip} = \Theta_b\dot{\phi}_b = const. \tag{7.16}$$

- conservation of angular momentum of full robot about prospective contact point

$$\begin{aligned} H_{robot,contact} &= \Theta_b\dot{\phi}_b - m_b(y_b - y_c)\dot{x}_b + m_b(x_b - x_c)\dot{y}_b \\ &\quad +\Theta_l\dot{\phi}_l - m_l(y_l - y_c)\dot{x}_l + m_l(x_l - x_c)\dot{y}_l = const. \end{aligned} \tag{7.17}$$

  with

$$x_c = x_b + (l_0 - r)\sin\phi_l \tag{7.18}$$

$$y_c = y_b - (l_0 - r)\cos\phi_l \tag{7.19}$$

- conservation of translational momentum in direction of leg (considering spring-damper-force)

$$m(\dot{x}_b\sin\phi_l - \dot{y}_b\cos\phi_l) - F_{kd} = const. \tag{7.20}$$

There is no discontinuity at liftoff.

The variable x_b describes the forward motion of the robot and is non-periodic. All other state variables have to satisfy periodicity constraints

$$\begin{pmatrix} y_b \\ \phi_b \\ \phi_l \\ \dot{x}_b \\ \dot{y}_b \\ \dot{\phi}_b \\ \dot{\phi}_l \end{pmatrix}(T) = \begin{pmatrix} y_b \\ \phi_b \\ \phi_l \\ \dot{x}_b \\ \dot{y}_b \\ \dot{\phi}_b \\ \dot{\phi}_l \end{pmatrix}(0) \tag{7.21}$$

where the period T is to be determined by the optimization.

7.2 Results of Stability Optimization

We present different stable solutions for one-legged hopping robots with circular and point foot. For both versions model parameters and trajectories with excellent stability properties were found. From a variety of cases computed we present the most important ones in this section.

We have applied the spectral radius as well as matrix norms for one and multiple steps as stability optimization criteria in the outer loop. A projection of the full monodromy matrix to the subspace of periodic variables had to be performed before. Ten out of twelve model parameters have been varied whereas leg length l_0 and torus mass m_b have been kept fixed for scaling reasons. Physically reasonable bounds have been imposed on all parameters. Specifically static instability of the system was thus maintained.

The objective function of Lagrange type applied in the inner loop was a sum of (weighted) controls squared. We have used a piecewise constant control discretization for both actuators. Control and multiple shooting grids each consist of 15 intervals per phase. State and control variables as well as phase times have to satisfy box constraints. Continuity of controls at dynamic discontinuity points has been guaranteed by equality constraints. Besides the periodicity constraints and switching functions described in section 7.1, we have imposed box constraints on all controls and states, a lower bound on the trunk forward speed at all points, and bounds on the leg inclination angle at touchdown and liftoff instants.

7.2.1 Point Foot

This is the first publication of results for a one-legged hopping robot with point foot that does not need feedback controllers but relies on open-loop stabilization instead.

Result of Eigenvalue Optimization

Using eigenvalue optimization we were able to bring the spectral radius down as far as 0.1292 for a one-legged hopping robot with point foot.

The model parameters of this solution are (in ISO units) $m_b = 2.0$, $\Theta_b = 0.3503$, $m_l = 0.5033$, $\Theta_l = 0.2391$, $d = 0.3663$, $l_0 = 0.5$, $r = 0$, $k_{tors} = 25.902$, $\Delta\phi_l = 0.2$, $b_{tors} = 3.457$, $k = 589.1$, and $b = 61.79$.

The initial values of the corresponding trajectory are

$$\begin{aligned} x_b(0) &= 0 & \dot{x}_b(0) &= 0.3326 \\ y_b(0) &= 0.490 & \dot{y}_b(0) &= 0.0011 \\ \phi_b(0) &= -0.1447 & \dot{\phi}_b(0) &= -2.8399 \\ \phi_l(0) &= 0.20 & \dot{\phi}_l(0) &= -0.6524 \end{aligned}$$

Figure 7.3 shows the control and state variable histories for this most stable solution. Bounds on all variables are represented as lines. The different phases – 1. contact phase, 2. flight phase, 3. touchdown transition phase of duration zero with velocity discontinuities – can be discerned in figure 7.3. Obviously all control variables and all state variables except x_b are periodic. If $x_b(0)$ is fixed to zero, $x_b(T)$ gives the step length of one hopping

cycle, in this case $0.536m$. The cycle time of this solution is $T = 0.471s$ with phase times $T_{contact} = 0.305s$ and $T_{flight} = 0.166s$.

Due to the non-periodicity of x_b, only seven out of eight eigenvalues are relevant for stability. This last eigenvalue, which is always one because of the system's indifference towards the initial value $x_b(0)$, is eliminated by projection. The seven relevant eigenvalues are

$$\begin{aligned} \lambda_{1,2} &= (0.1088, \pm 0.0696) \\ \lambda_{3,4} &= (-0.0888, \pm 0.0913) \\ \lambda_5 &= -0.0722 \\ \lambda_{6,7} &= 0 \end{aligned}$$

and by magnitude

$$\begin{aligned} |\lambda_{1,2}| &= 0.1292 \\ |\lambda_{3,4}| &= 0.1274 \\ |\lambda_5| &= 0.072172 \\ |\lambda_{6,7}| &= 0. \end{aligned}$$

The two eigenvalues of zero magnitude are caused by the reduction from four to three DOFs during contact phase and the resulting coupling of perturbations in velocity as well as position space.

For this reduced monodromy matrix we have the following matrix norms

$$\begin{aligned} \sigma_{max} &= 12.1911 \\ ||C||_\infty &= 16.7044 \\ ||C||_1 &= 23.1978. \end{aligned}$$

Perturbations therefore don't contract in any of these norms over one step. But as a study of matrix powers shows (figure 7.4), they all do contract over cycles of four and more steps.

With the spectral radius being far below one we have proven stability according to linear theory. But its size does not say anything about the size of perturbations from which the system can recover. We determine these stability margins according to the procedure described in section 6.7. The robot can recover from substantial perturbations of its initial values under the invariant influence of its periodic actuations:

ϕ_b	+ 133%	-63%
y_b, ϕ_l	−3%, +57%	+0.6%, −17%
$\dot{x}_b$	+39%	-90%
$\dot{y}_b$	+5000%	-100%
$\dot{\phi}_b$	+23%	-42%
$\dot{\phi}_l$	+27%	-46%

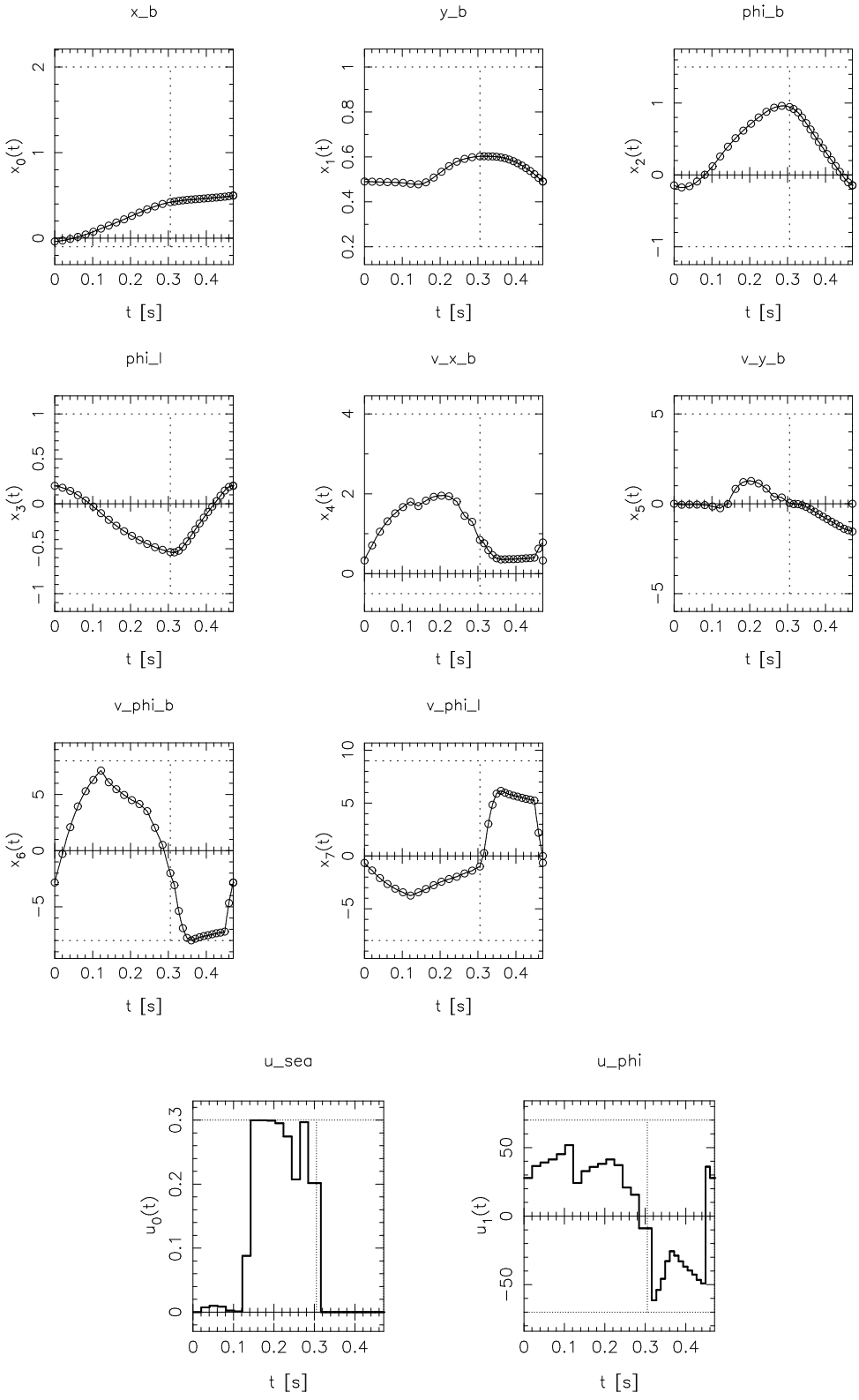


Figure 7.3: State and control variable trajectories of most stable solution for Hopper with point foot

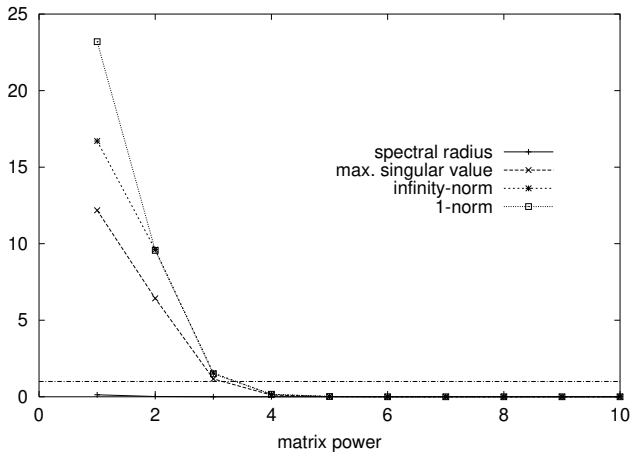


Figure 7.4: Spectral radius versus matrix norms for different matrix powers

For the non-periodic variable x_b of course arbitrary initial values can be chosen. As indicated by the brace, y_b and ϕ_l are perturbed together such that the initial values remain consistent with the touchdown manifold. Figure 7.5 illustrates the differences between the original periodic trajectory and one for which the initial value of $\dot{x}_b$ has been perturbed by -90%. Obviously the robot stays synchronized with its exciting frequency. The perturbed trajectory is characterized by shorter steplengths, i.e. it stays behind the base solution in the non-periodic variable x_b.

The robot also persists in its hopping motion under the following perturbations of model parameter values:

Θ_b	$+5\%$	-1%
m_l	$+5\%$	-20%
Θ_l	$+4\%$	-23%
d	$+11\%$	-37&
k_{tors}	$+3\%$	-9%
$\Delta\phi$	$+96\%$	-45%
b_{tors}	$+1\%$	-5%
k	$+1\%$	-0.4%
b	$+0.5\%$	-2%

Result of Singular Value Optimization

We were interested in finding out

- if a solution with maximum singular value smaller than one existed for the one-legged hopping robot

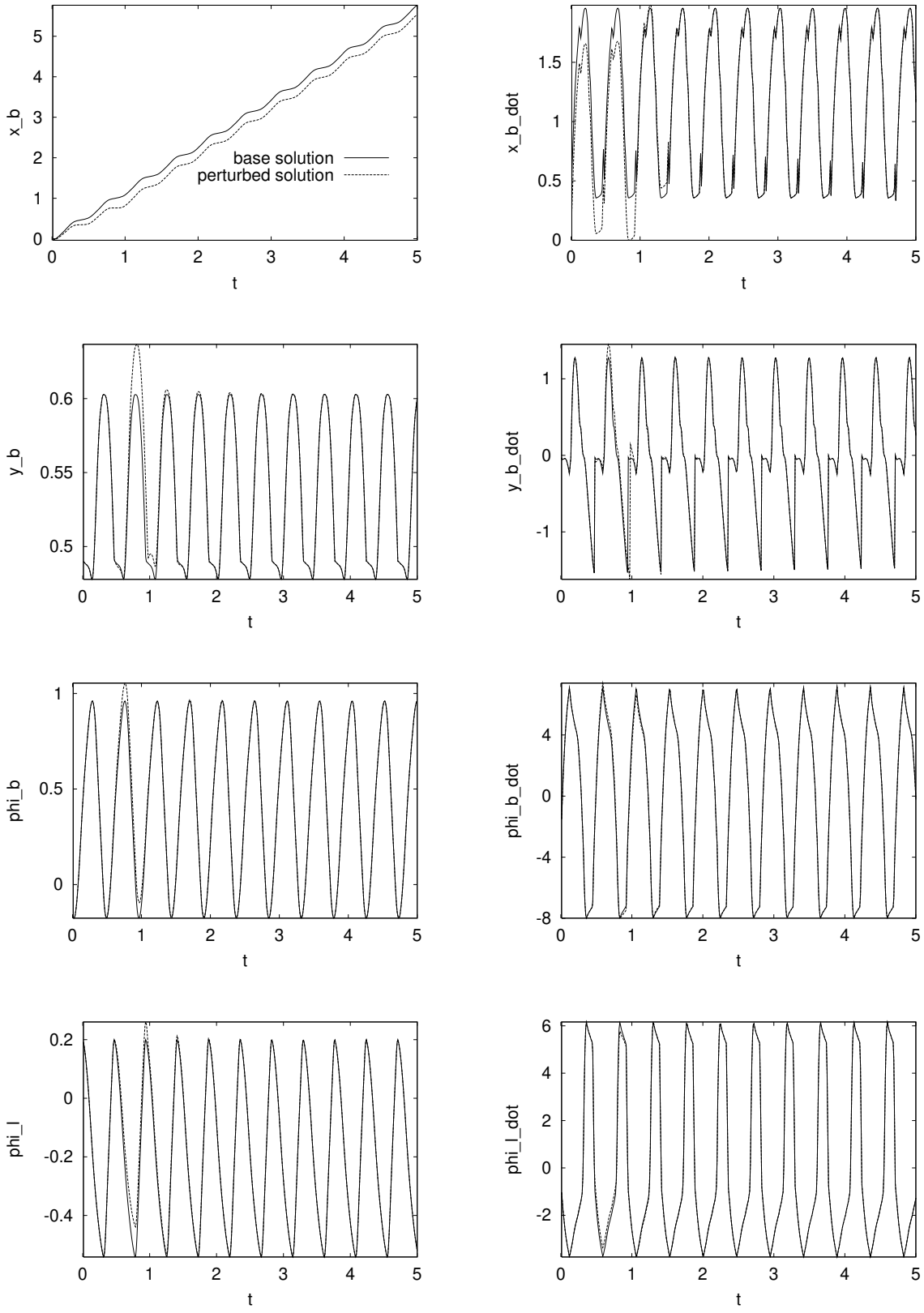


Figure 7.5: Most stable periodic solution for point foot hopper with and without perturbation (in $\dot{x}_b$) in all position (left) and velocity (right) variables

- if singular value optimization could help to find stable solutions probably even better than the above solution.

The answer to both questions is no. Singular value optimization of the monodromy matrix produced a solution characterized by a maximum singular value of

$$\sigma_{max}(C) = 3.511,$$

smaller than for the previous solution, but by far larger than one. The maximum eigenvalue of the resulting matrix is, by magnitude

$$|\lambda_{max}(C)| = 1.383,$$

and the system is therefore unstable. Figure 7.6 illustrates the development of the spectral radius during singular value optimization. While the maximum singular value is decreased, the spectral radius even deteriorates during optimization.

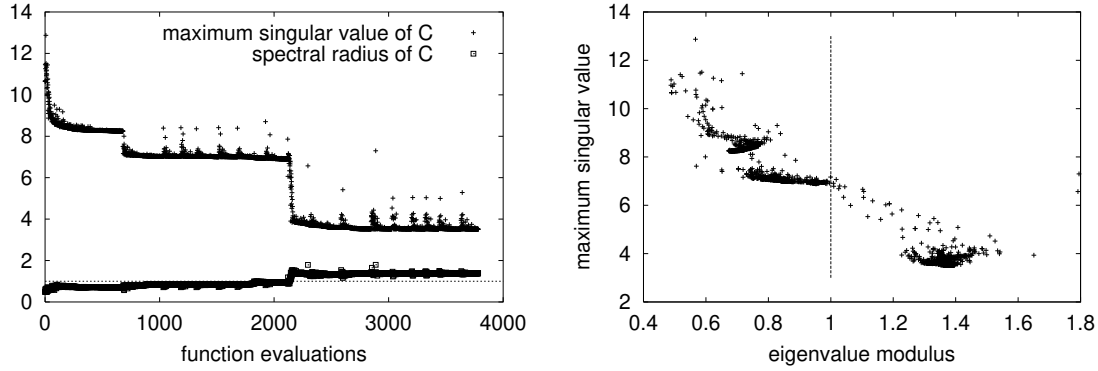


Figure 7.6: Development of spectral radius during optimization of maximum singular value of monodromy matrix

The model parameters of this solution are $m_b = 2.0$, $\Theta_b = 0.995$, $m_l = 0.9618$, $\Theta_l = 0.3$, $d = 0.1752$, $l_0 = 0.5$, $r = 0$, $k_{tors} = 17.84$, $\Delta\phi_l = 0.2$, $b_{tors} = 5.649$, $k = 287.09$, and $b = 39.74$. It has the initial values

$$x_0^T = (-0.0386, 0.490, -0.1447, 0.2, 0.2, -0.0835, -1.083, -0.359)$$

and a period of $T = 0.504s$ with phase durations $T_{contact} = 0.326s$ and $T_{flight} = 0.178s$.

We conclude that the maximum singular value of the monodromy matrix is not a favorable optimization criterion for the present case.

Result of Singular Value Optimization of Matrix Power

In section 5.3 we have discussed the use of a norm of a power of the matrix instead of the matrix itself as stability optimization criterion. Here we have chosen to apply the singular value of the fourth matrix power.

It serves the purpose much better than the singular value of the matrix itself. The optimization resulted in a singular value of

$$\sigma_{max}(C^4) = 0.0756$$

and maximum eigenvalues of

$$\begin{aligned} |\lambda_{max}(C^4)| &= 0.00116 \quad \text{and} \\ |\lambda_{max}(C)| &= 0.1848. \end{aligned}$$

The solution is therefore stable. Figure 7.7 shows the course of eigenvalues of the monodromy matrix and its fourth power during singular value optimization. The present

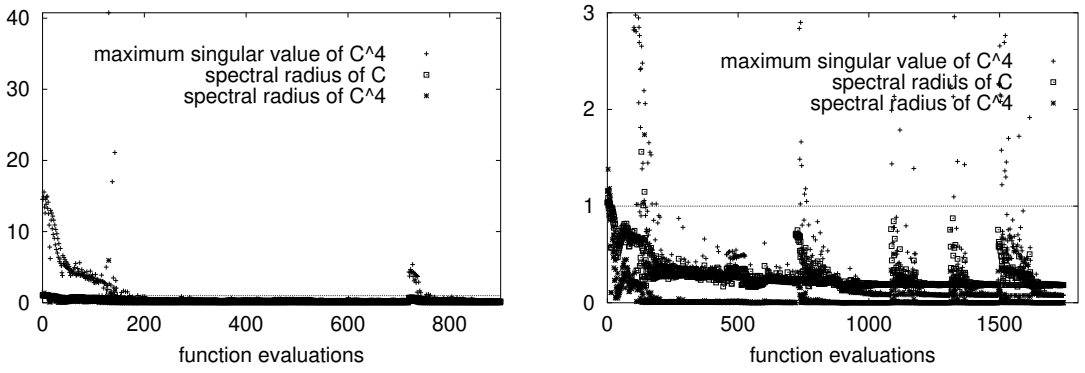


Figure 7.7: Development of spectral radius during optimization of maximum singular value of fourth matrix power

optimization criterion is obviously successful in decreasing the spectral radius. It even produces an absolute value not very far above the one obtained by eigenvalue optimization.

The solution under discussion is characterized by model parameters $m_b = 2.0$, $\Theta_b = 0.3510$, $m_l = 0.5031$, $\Theta_l = 0.2395$, $d = 0.3666$, $l_0 = 0.5$, $r = 0$, $k_{tors} = 25.90$, $\Delta\phi_l = 0.2$, $b_{tors} = 3.456$, $k = 588.86$, and $b = 60.847$, initial values

$$x_0^T = (-0.0385, 0.490, -0.1447, 0.2005, 0.3299, -0.00048, -2.849, -0.6463)$$

and a period of $T = 0.4718s$ with phase times $T_{contact} = 0.3056s$ and $T_{flight} = 0.1662s$.

7.2.2 Circular Foot

Open-loop controlled hopping robots with circular foot were discovered before but they generally rely on a large foot radius for stabilization (Ringrose [77], Wei et al. [99]). We enforce a center of mass position above the centers of foot curvature ($l_0 - d > r$) and thus static instability of the robot during optimization.

Stable results for a hopping robot with a small circular foot have been published in Mombaur et al. [63]. The solutions presented here are characterized by even better maximum eigenvalues.

Result of Eigenvalue Optimization

In order to guarantee a clearly circular foot shape, we have restricted our search to $r > 0.1m$. The most stable solution found under this condition has a spectral radius of 0.2872.

It is quite similar to the result of eigenvalue optimization for a robot with point foot. The model parameter values of the solution are $m_b = 2.0$, $\Theta_b = 0.2385$, $m_l = 0.5078$, $\Theta_l = 0.2468$, $d = 0.3$, $l_0 = 0.5$, $r = 0.1021$, $k_{tors} = 25.492$, $\Delta\phi_l = -0.1259$, $b_{tors} = 2.443$, $k = 555.30$, and $b = 58.178$.

Also the control variable histories and trajectories as shown in figure 7.8 are only slightly different from the point foot solution The corresponding cycle time is $T = 0.478s$ with $T_{contact} = 0.301s$ and $T_{flight} = 0.177s$ and the initial values

$$x_0^T = (-0.0385, 0.4921, -0.1447, 0.2, 0.6370, 0.0547, -1.307, -1.597)$$

It is traveling faster in x_b-direction and has a larger step length of $0.6032m$.

The related monodromy matrix has the following eigenvalues:

$$\begin{aligned} \lambda_{1,2} &= (-0.0224, \pm 0.2761) & |\lambda_{1,2}| &= 0.2770 \\ \lambda_3 &= 0.2872 & |\lambda_3| &= 0.2872 \\ \lambda_4 &= 0.2710 & |\lambda_4| &= 0.2710 \\ \lambda_5 &= -0.0276 & |\lambda_5| &= 0.0276 \\ \lambda_{6,7} &= 0 & |\lambda_{6,7}| &= 0. \end{aligned}$$

Again, the matrix norms describing the propagation of perturbations over one step are all larger than one,

$$\begin{aligned} \sigma_{max} &= 17.869 \\ ||C||_\infty &= 24.466 \\ ||C||_1 &= 27.063. \end{aligned}$$

but if plotted as functions of the matrix power, the norms are contractive for exponents greater or equal to five (figure 7.9).

The region of stability in which the robot can recover and maintain a gait without falling down is described by the stability margins

ϕ_b	+135%	-245%
y_b }	−0.48%	+0.75%
ϕ_l }	+14%	−25%
$\dot{x}_b$	+21%	-11%
$\dot{y}_b$	+130%	-99%
$\dot{\phi}_b$	+92%	-57%
$\dot{\phi}_l$	+15%	-14%.

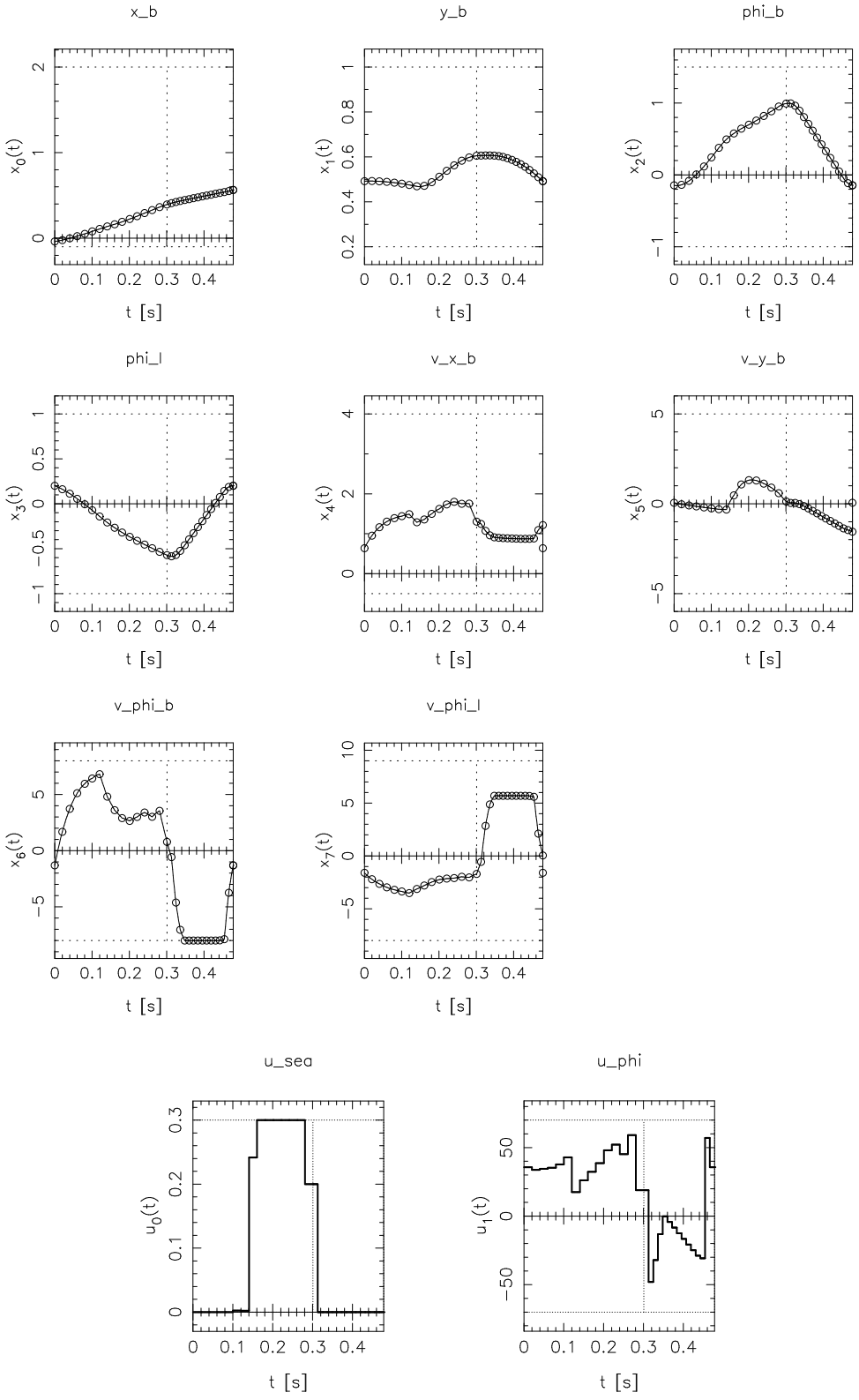


Figure 7.8: State and control variable histories of most stable solution for Hopper with circular foot

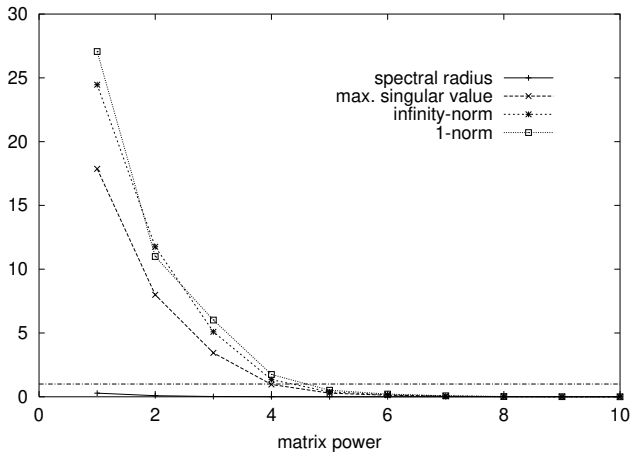


Figure 7.9: Spectral radius versus matrix norms for different matrix powers

In figure 7.10 we compare the trajectory starting from a perturbed value of ϕ_b (+135%) with the corresponding unperturbed solution.

The following table finally lists the maximum possible perturbations of model parameters:

Θ_b	+0.3%	-1%
m_l	+4%	-1%
Θ_l	+2%	-1%
d	+12%	-2%
r	+14%	-0.3%
k_{tors}	+1%	-0.3%
$\Delta\phi$	+280%	-160%
b_{tors}	+1%	-0.3%
k	+0.2%	-0.8%
b	+3%	-1%

Result of ∞-norm Optimization

Again we aimed at finding a solution with a contracting norm of the monodromy matrix, this time using the infinity norm as optimization criterion. But the effect is the same as encountered for singular value optimization of the point foot model: The minimum value found is

$$||C||_\infty = 3.744,$$

significantly smaller than for the previous solution, but not smaller than one as desired. The corresponding maximum eigenvalue has a magnitude of

$$|\lambda_{max}(C)| = 1.627 > 1$$

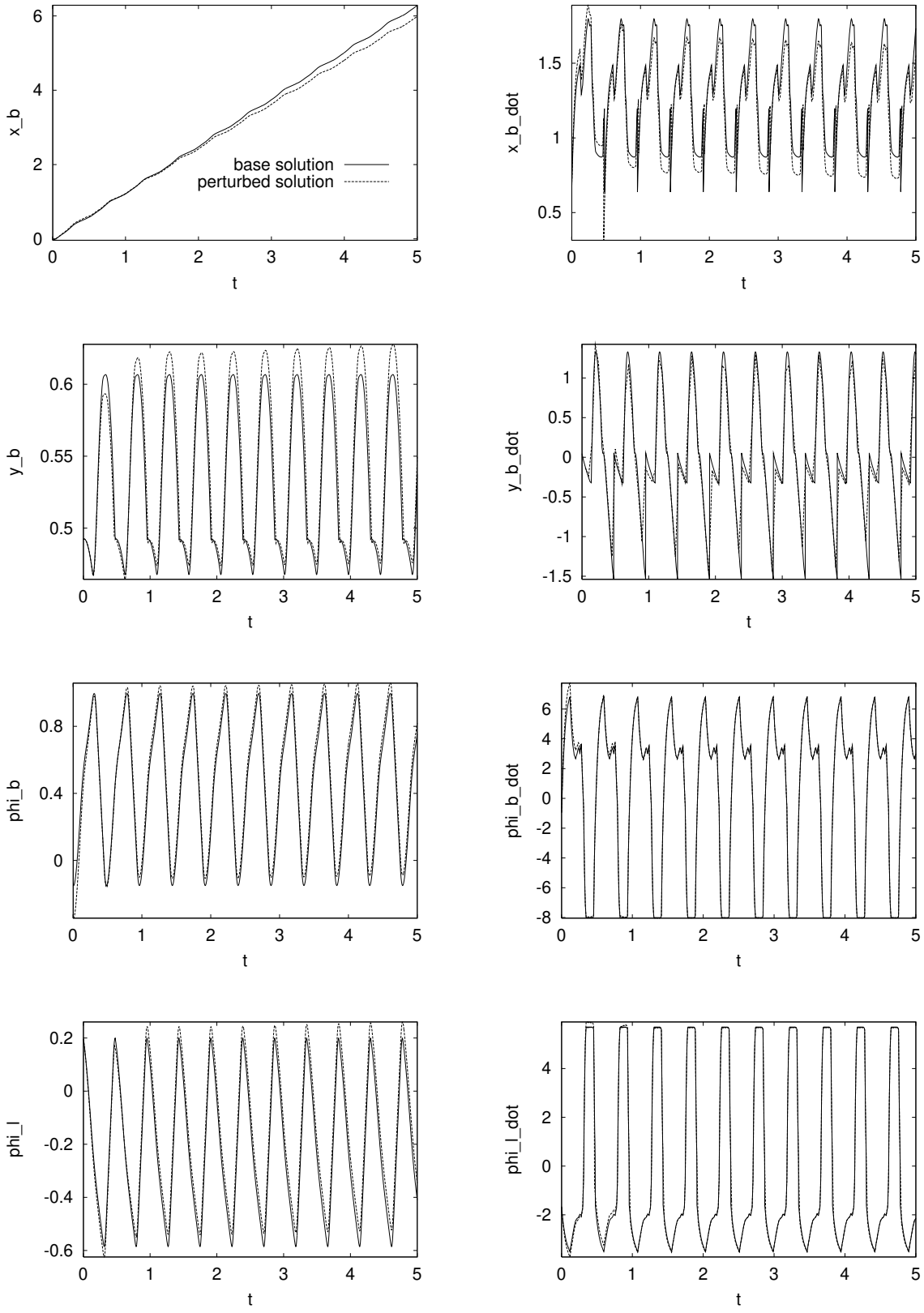


Figure 7.10: Most stable periodic solution for circular foot hopper with and without perturbation (in ϕ_b) in all position (left) and velocity (right) variables

The solution is thus unstable and not of practical relevance. As depicted in figure 7.11, the spectral radius is growing, while the infinty norm decreases.

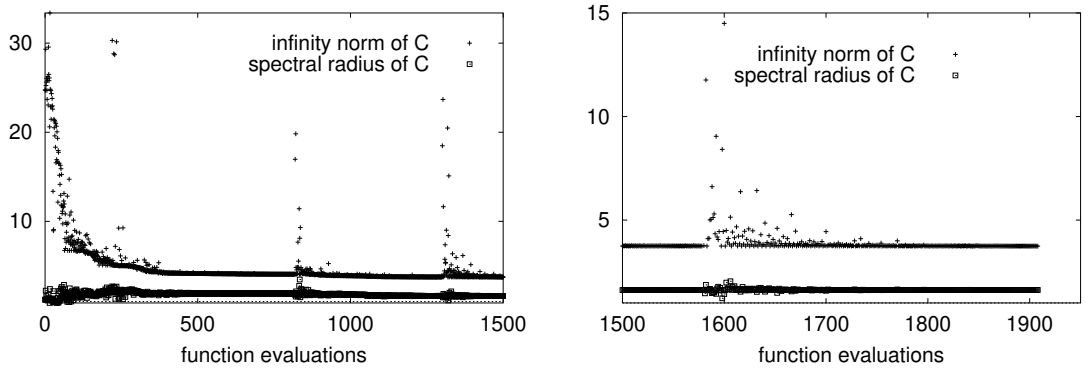


Figure 7.11: Development of eigenvalue during optimization of infinity norm of monodromy matrix

The solution is characterized by parameter values of $m_b = 2.0$, $\Theta_b = 0.2404$, $m_l = 1.495$, $\Theta_l = 0.2880$, $d = 0.1390$, $l_0 = 0.5$, $r = 0.1270$, $k_{tors} = 25.39$, $\Delta\phi_l = 0.0418$, $b_{tors} = 2.819$, $k = 453.79$, and $b = 64.67$, a trajectory starting at

$$x_0^T = (-0.0385, 0.4926, -0.1447, 0.2, 0.2545, -0.0469, -2.832, -0.6436)$$

and phase times $T_{contact} = 0.3451s$ and $T_{flight} = 0.1869s$, leading to a period of $T = 0.532s$.

Result of ∞-norm Optimization of Matrix Power

Using the infinity norm of a power of the monodromy matrix instead (here the fourth power), we found a stable solution. The optimal value is

$$|C^4|_\infty = 0.3306,$$

i.e. measured in the ∞-norm, perturbations decay to about a third of their original size over a cycle of four steps.

The corresponding maximum eigenvalues are

$$\begin{aligned} |\lambda_{max}(C^4)| &= 0.0161 \quad \text{and} \\ |\lambda_{max}(C)| &= 0.3560 \end{aligned}$$

characterizing a stable solution. This confirms the observation made in section 7.2.2, that the norm of a matrix power should be preferred as optimization criterion over a norm of the monodromy matrix itself. The solution has similar properties as the result of eigenvalue optimization and only a slightly larger spectral radius.

The model parameters of this solution are $m_b = 2.0$, $\Theta_b = 0.2463$, $m_l = 0.5177$, $\Theta_l = 0.2405$, $d = 0.3$, $l_0 = 0.5$, $r = 0.1022$, $k_{tors} = 25.498$, $\Delta\phi_l = -0.0913$, $b_{tors} = 2.387$, $k = 560.61$, and $b = 58.437$. Its trajectory has the initial values

$$x_0^T = (-0.0385, 0.4909, -0.1447, 0.2146, 0.6276, 0.0617, -1.4423, -1.574)$$

and a cycle time of $T = 0.481s$ with phase times $T_{contact} = 0.3038s$ and $T_{flight} = 0.1772s$.

7.3 Summary

We summarize the most important results of this chapter.

- New open-loop stable trajectories for one-legged hopping robots with point feet and circular feet have been presented. To our knowledge this is the first report about a one-legged point foot robot that is stable without feedback controllers. Hopping robots with circular feet did exist before, but we reduced the foot radius thus not allowing for trivial stabilizing effects.
- We stress the fact that a circular foot is not necessary for open-loop stability of one-legged hopping robots.
- All our robots intentionally have no statically stable standing configurations as also for the version with circular feet the centers of mass lie above the centers of foot curvature ($l_0 - d > r$).
- The solutions presented exhibit excellent linear and nonlinear stability properties. They are not only characterized by very small spectral radii of their monodromy matrices but they also can sustain substantial perturbations of the initial values and model parameters.
- No solution has been found for which an induced matrix norm of the monodromy matrix over one step is smaller than one. But we report several solutions for which perturbations do contract in the 1-, 2-, or ∞-norm over a cycle of several steps.
- From the previous two items we can conclude that the existence of a contractive norm of the monodromy matrix is not a necessary condition for excellent stability of a solution in the nonlinear sense.
- The usage of a norm of the monodromy matrix as optimization criterion had no favorable influence on the eigenvalues in the cases tested: the spectral radius very often deteriorated during the course of optimization and was larger than one at the convergence point.
- Using the norm of a power of the monodromy matrix proved to be a much better choice. Both the ∞-norm and the maximum singular value of a power served to bring the eigenvalue down below one and led to a solution that was very close to the solution found with eigenvalue optimization.

Chapter 8

Open-loop Stable Human-like Actuated Walking Robot

In this chapter we study a two-legged kneed walking robot with point feet. The robot can be considered as a simplified model of human walking in the saggital plane. It is powered by periodic torque actuations at hip and knee that are not changed by feedback interference. Nevertheless the robot is capable of naturally recovering from perturbations. We believe that our robot is the first demonstration of a human-like actuated open-loop stable gait.

The robot consists of four bodies – two symmetric legs with a thigh and a shank each. We have not added a trunk as already this simpler mechanism exhibits a remarkably human-like gait (see the animation sequence 8.2). The completion of the model by a trunk is an easy task but would not provide any further insights with respect to the objective of this thesis which is to find open-loop stable robots.

Inelastic ground collisions cause the system to be non-conservative. The motion is piece-wise holonomic but overall non-holonomic. According to section 1.4 both properties can help to increase stability of the robot.

In this thesis we extend the results recently published in Mombaur et al. [64]. We study the effects of different objective functions for stability optimization and give further stable solutions.

There is an equivalent passive-dynamic version of this robot which has no actuators but walks on an inclined slope instead. It is similar to the passive dynamic walkers of McGeer and the Ruina lab presented in section 1.3 except that it doesn't have circular feet. We have studied this passive robot earlier and published the results in Mombaur et al. [62]. They are used here for comparison with the results of the actuated robot.

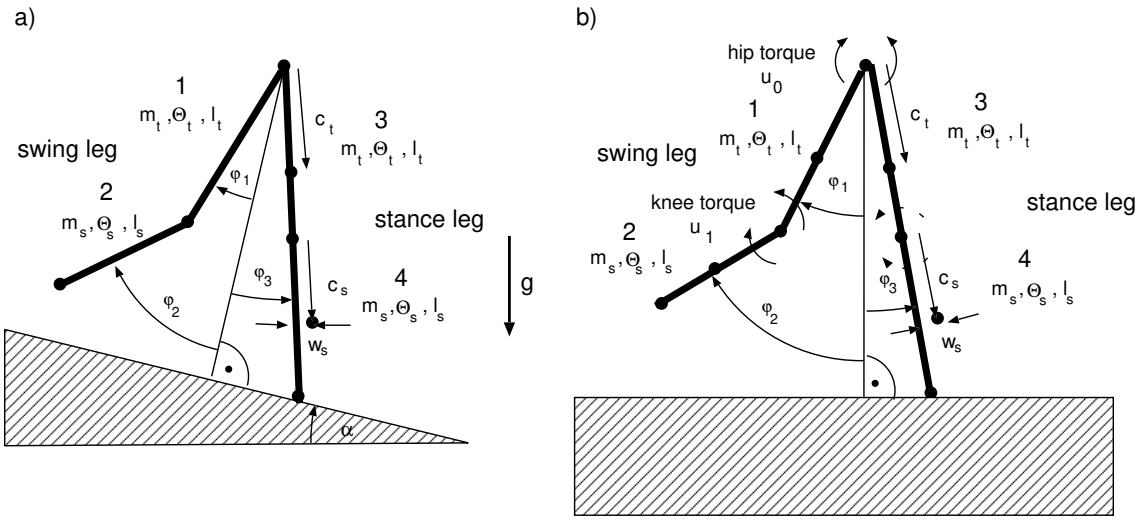


Figure 8.1: Parameters and configuration variables of passive (a) and active (b) kneed walker

8.1 Robot Model

For the sake of completeness, we derive the model equations for both the actuated and the passive walking robot. Figure 8.1 illustrates model parameters and geometric variables for both versions. Seven parameters are used for the actuated model: masses m_i, lengths l_T, and center of mass locations c_i, w_i of thigh and shank ($i = T, S$, $w_T = 0$). For the passive version, the slope angle α is an additional parameter. The moments of inertia of thigh and shank can be computed from these quantities:

$$\Theta_i = \frac{1}{6} m_i (l_i^2 + 2c_i^2 - 2l_i c_i). \tag{8.1}$$

We assume that ground contact occurs without sliding and and that there is no double-support phase, i. e. the second leg instantaneously leaves the ground after heelstrike. This is a very common assumption for the simulation of walking motions (compare e.g. Channon et al. [14], McGeer [52]).

We model one step - and not a full physical cycle consisting of two steps - because we are only interested in symmetric gaits. The observed cycle starts and ends right after heelstrike. The stance leg is assumed to be straight all the time, whereas the swing leg is bent in the first phase and straight in the second phase after kneestrike, such that the robot has three or two degrees of freedom, respectively. For both phases, we use the uniform set of optimization coordinates $q = (\phi_1, \phi_2, \phi_3)^T$ (angles of swing leg thigh and shank and of total stance leg – as $\phi_4 \equiv \phi_3$) and the corresponding rates $\dot{q} = (\dot{\phi}_1, \dot{\phi}_2, \dot{\phi}_3)^T$ for all phases. They are minimal coordinates for the first motion phase and redundant coordinates for the second phase. Note again that due to the overall non-holonomy of the gait, this set of coordinates would not be sufficient to describe a multi-step motion.

The equations of motion have been derived using free-body-diagrams (section 2.2.2). In

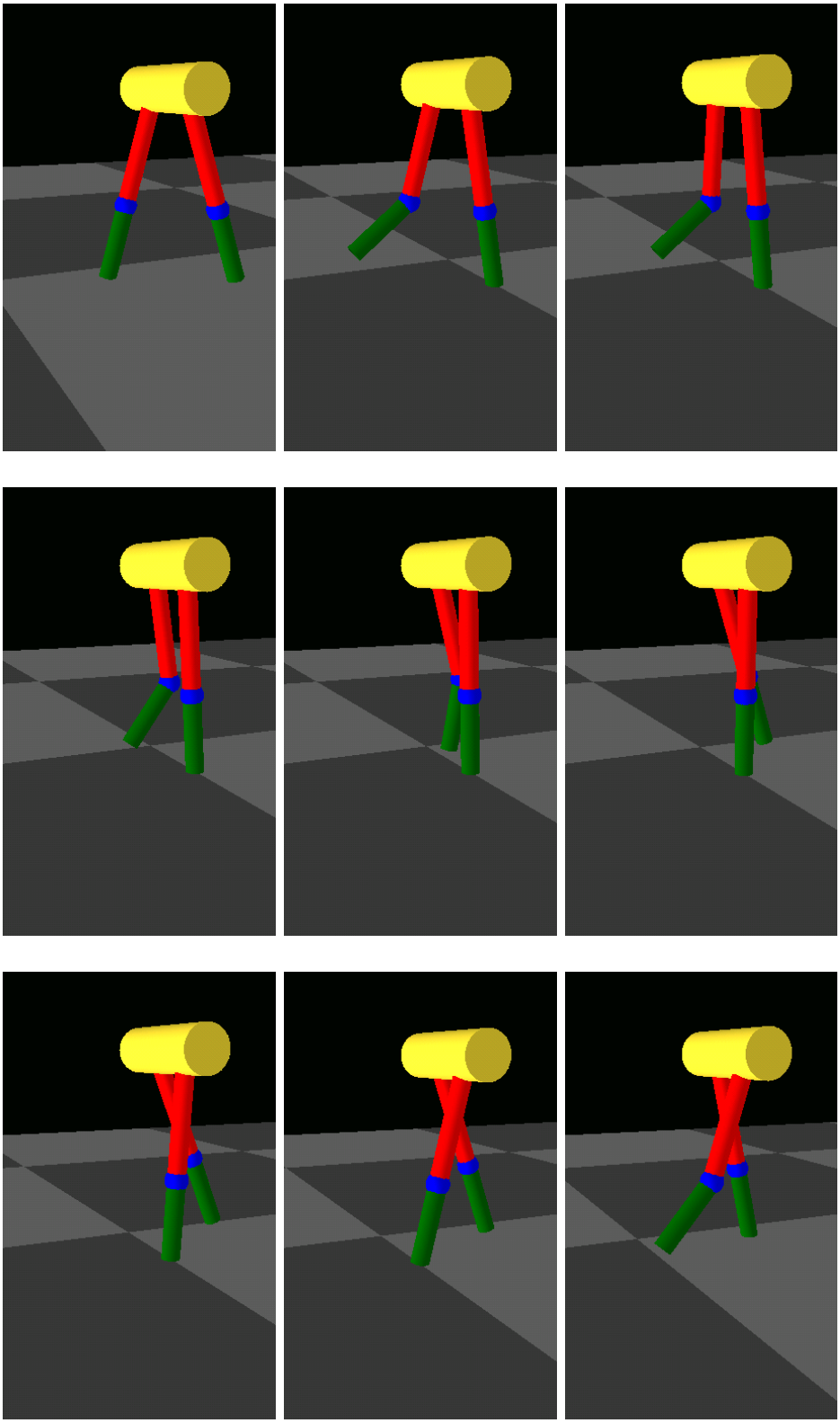


Figure 8.2: Motion of bipedal walking robot

this context, the position variables of all centers of mass r_i and of the hip r_H as well as ϕ_4 have been used as auxiliary coordinates. Size and complexity of the system forbids to explicitely solve the system for the optimization coordinates. Instead the following set of 23/24 equations is solved numerically in each step:

a) Newton's laws of motion in translational and rotational direction for individual bodies:

$$\begin{aligned}
m_T\ddot{r}_{x,1} &= F_{2x} - F_{1x} + m_T g\sin\alpha && (8.2)\\
m_T\ddot{r}_{y,1} &= F_{2y} - F_{1y} - m_T g\cos\alpha && (8.3)\\
\theta_T\ddot{\phi}_1 &= (F_{1x}c_T + F_{2x}(l_T - c_T))\cos\phi_1 \\
&\quad +(F_{1y}c_T + F_{2y}(l_T - c_T))\sin\phi_1 + u_1 - u_0 && (8.4)
\end{aligned}$$

$$\begin{aligned}
m_S\ddot{r}_{x,2} &= -F_{2x} + m_S g\sin\alpha && (8.5)\\
m_S\ddot{r}_{y,2} &= -F_{2y} - m_S g\cos\alpha && (8.6)\\
\theta_S\ddot{\phi}_2 &= F_{2x}(c_S\cos\phi_2 - w_S\sin\phi_2) + F_{2y}(c_S\sin\phi_2 + w_S\cos\phi_2) - u_1 && (8.7)
\end{aligned}$$

$$\begin{aligned}
m_T\ddot{r}_{x,3} &= F_{3x} + F_{1x} + m_T g\sin\alpha && (8.8)\\
m_T\ddot{r}_{y,3} &= F_{3y} + F_{1y} - m_T g\cos\alpha && (8.9)\\
\theta_T\ddot{\phi}_3 &= (-F_{1x}c_T + F_{3x}(l_T - c_T))\cos\phi_3 \\
&\quad +(-F_{1y}c_T + F_{3y}(l_T - c_T))\sin\phi_3 + u_0 + u_2 && (8.10)
\end{aligned}$$

$$\begin{aligned}
m_S\ddot{r}_{x,4} &= -F_{3x} + B_x + m_S g\sin\alpha && (8.11)\\
m_S\ddot{r}_{y,4} &= -F_{3y} + B_y - m_S g\cos\alpha && (8.12)\\
\theta_S\ddot{\phi}_4 &= F_{3x}(c_S\cos\phi_4 - w_S\sin\phi_4) + F_{3y}(c_S\sin\phi_4 + w_S\cos\phi_4) \\
&\quad +B_x((l_S - c_S)\cos\phi_3 + w_S\sin\phi_3) + B_y((l_S - c_S)\sin\phi_3 \\
&\quad -w_S\cos\phi_3) - u_2 && (8.13)
\end{aligned}$$

with $u_2 \equiv 0$. F_{jx} and F_{jy}, $(j = 1, 2, 3)$ are constraint forces at hip joint, swing leg knee and stance leg knee and B_x and B_y ground reaction forces. For the passive version, $u_0 \equiv u_1 \equiv 0$. For the actuated version, $\alpha = 0$, and $u_1 \equiv 0$ during the second phase.

b) Kinematic equations:

$$\begin{aligned}
\ddot{r}_{x,1} &= \ddot{r}_{x,H} + c_T(\cos\phi_1\ddot{\phi}_1 - \sin\phi_1\dot{\phi}_1^2) && (8.14)\\
\ddot{r}_{y,1} &= \ddot{r}_{y,H} + c_T(\sin\phi_1\ddot{\phi}_1 + \cos\phi_1\dot{\phi}_1^2) && (8.15)
\end{aligned}$$

$$\begin{aligned}
\ddot{r}_{x,2} &= \ddot{r}_{x,H} + l_T(\cos\phi_1\ddot{\phi}_1 - \sin\phi_1\dot{\phi}_1^2) \\
&\quad +c_S(\cos\phi_2\ddot{\phi}_2 - \sin\phi_2\dot{\phi}_2^2) - w_S(\sin\phi_2\ddot{\phi}_2 + \cos\phi_2\dot{\phi}_2^2) && (8.16)\\
\ddot{r}_{y,2} &= \ddot{r}_{y,H} + l_T(\sin\phi_1\ddot{\phi}_1 + \cos\phi_1\dot{\phi}_1^2)
\end{aligned}$$

$$+c_S(\sin\phi_2\ddot{\phi}_2+\cos\phi_2\dot{\phi}_2^2)+w_S(\cos\phi_2\ddot{\phi}_2-\sin\phi_2\dot{\phi}_2^2) \tag{8.17}$$

$$\ddot{r}_{x,3} = \ddot{r}_{x,H}+c_T(+\cos\phi_3\ddot{\phi}_3-\sin\phi_3\dot{\phi}_3^2) \tag{8.18}$$
$$\ddot{r}_{y,3} = \ddot{r}_{y,H}+c_T(+\sin\phi_3\ddot{\phi}_3+\cos\phi_3\dot{\phi}_3^2) \tag{8.19}$$

$$\begin{aligned}\ddot{r}_{x,4} &= \ddot{r}_{x,H}+l_T(\cos\phi_3\ddot{\phi}_3-\sin\phi_3\dot{\phi}_3^2)\\ &\quad +c_S(\cos\phi_4\ddot{\phi}_4-\sin\phi_4\dot{\phi}_4^2)-w_S(\sin\phi_4\ddot{\phi}_4+\cos\phi_4\dot{\phi}_4^2) \qquad (8.20)\\ \ddot{r}_{y,4} &= \ddot{r}_{y,H}+l_T(\sin\phi_3\ddot{\phi}_3+\cos\phi_3\dot{\phi}_3^2)\\ &\quad +c_S(\sin\phi_4\ddot{\phi}_4+\cos\phi_4\dot{\phi}_4^2)+w_S(\cos\phi_4\ddot{\phi}_4-\sin\phi_4\dot{\phi}_4^2) \qquad (8.21)\end{aligned}$$

$$\phi_4 \equiv \phi_3 \tag{8.22}$$
$$\dot{\phi}_4 \equiv \dot{\phi}_3 \tag{8.23}$$
$$\ddot{\phi}_4 \equiv \ddot{\phi}_3 \tag{8.24}$$

$$(\ddot{\phi}_2 \equiv \ddot{\phi}_1) \tag{8.25}$$

The last equation is only valid during the second phase.

Kneestrike and heelstrike are modeled as perfectly inelastic impacts which result in velocity discontinuities and energy dissipation.

Kneestrike occurs when the relative angle between thigh and shank is zero:

$$s_{kneestrike}(x)=\phi_1-\phi_2=0. \tag{8.26}$$

Additionally, the rate of the shank needs to be larger than that of the thigh in order to cause impact:

$$c_{kneestrike}(x)=\dot{\phi}_1-\dot{\phi}_2>0. \tag{8.27}$$

With the assumption that exciting torques for the actuated robot version are continuous at kneestrike, velocity jumps for both model versions are uniquely determined by the conditions

- conservation of angular momentum of swing leg about hip point H

$$H_{swing,H}-M_{Hip}=\sum_{i=1}^{2}\left(r_{H,i}\times m\dot{r}_i+\Theta_{i,z}\dot{\phi}_i\right)=const. \tag{8.28}$$

- conservation of angular momentum of robot about stance point S

$$H_{robot,S}=\sum_{i=1}^{4}\left(r_{S,i}\times m\dot{r}_i+\Theta_{i,z}\dot{\phi}_i\right)=const. \tag{8.29}$$

- equal angular velocities of thigh and shank of swing leg after the impact:

$$\dot{\phi}_1 = \dot{\phi}_2. \tag{8.30}$$

For non-continuous torques, the hip torque difference would have to be included in the first balance.

Heelstrike takes place when the height of the swing leg heel approaches zero:

$$s_{heelstrike}(x,p) = (l_T + l_S)\cos\phi_3 - l_T\cos\phi_1 - l_S\cos\phi_2. \tag{8.31}$$

The vertical velocity of the heel at this point has to be negative

$$c_{heelstrike}(x,p) = -(l_T + l_S)\sin\phi_3\dot{\phi}_3 + l_T\sin\phi_1\dot{\phi}_1 + l_S\sin\phi_2\dot{\phi}_2 < 0 \tag{8.32}$$

The heelstrike transition phase also includes a shifting of legs. The swing leg becomes the stance leg and vice versa. which causes the change of indices in the equations below.

At heelstrike, the assumptions of no impulse on the former stance leg when leaving the ground but only on the former swing leg when hitting the ground (see section 2.1.3) and of continuous torques lead to the following set of conditions: Conservation of angular momentum

- of whole robot about new contact point D

$$H_{robot,D} = \sum_{i=1}^{4}\left(r_{D,i}\times m\dot{r}_i + \Theta_{i,z}\dot{\phi}_i\right) = const. \tag{8.33}$$

- of former stance leg about hip point H

$$\begin{aligned}(H^{-}_{stance,hip} =)\ \sum_{i=3}^{4}\left(r_{H,i}\times m\dot{r}_i + \Theta_{i,z}\dot{\phi}_i\right) &= \\ (H^{+}_{swing,hip} =)\ \sum_{i=1}^{2}\left(r_{H,i}\times m\dot{r}_i + \Theta_{i,z}\dot{\phi}_i\right)&\end{aligned} \tag{8.34}$$

- and of former stance shaft about knee

$$\begin{aligned}(H^{-}_{stance-shaft,knee} =)\ r_{H,4}\times m\dot{r}_4 + \Theta_{4,z}\dot{\phi}_4 &= \\ (H^{+}_{swing-shaft,knee} =)\ r_{H,2}\times m\dot{r}_2 + \Theta_{4,z}\dot{\phi}_2&\end{aligned} \tag{8.35}$$

must be guaranteed. Again, for non-continuous torques, values before and after collision would have to be taken into account for the last two balance equations.

The inclusion of the leg shift in the transition phase allows us to apply periodicity constraints to all positions and velocities after one step of time T (to be determined):

$$\begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \end{pmatrix} (T) = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \end{pmatrix} (0) \tag{8.36}$$

8.2 Results of Stability Optimization

In this section we present our key optimization results for the kneed walking robot. Several performance criteria have been evaluated: the spectral radius of the monodromy matrix as well as induced norms of this matrix and its powers. All seven model parameters are varied in the outer stability optimization loop. We also have studied the size of stable regions in the neighborhood of the solutions found. We compare the results of nonlinear stability analysis with the corresponding passive solutions.

In the inner loop optimal control problem we have again minimized a sum of weighted torques squared. They have been discretized as piecewise constant functions on a grid with ten intervals per phase. The same grid was used for multiple shooting. Knee flexion during the first phase, clearance of the swing foot during the full step and a minimum leg inclination at the initial point have been enforced by inequality constraints at the respective multiple shooting nodes. Box constraints have been imposed on state and control variables and phase times. Periodicity and switching conditions have been formulated as coupled and decoupled equality constraints.

Result of eigenvalue optimization

The most stable solution for the actuated kneed walking robot found by eigenvalue optimization has a spectral radius of 0.5667. The solution was recently reported in Mombaur et al. [64]. We give further details in this section.

Figure 8.3a shows the trajectory describing one step with its two phases of motion. Discontinuities occur kneestrike and heelstrike in the middle and at the end respectively. Since the final discontinuity also includes a leg shift, not only the velocities but also the position variables are discontinuous.

The corresponding actuator torques are shown in figure 8.3b. Note the continuity of controls at the discontinuies of the state variables.

The initial values of this most stable periodic trajectory are

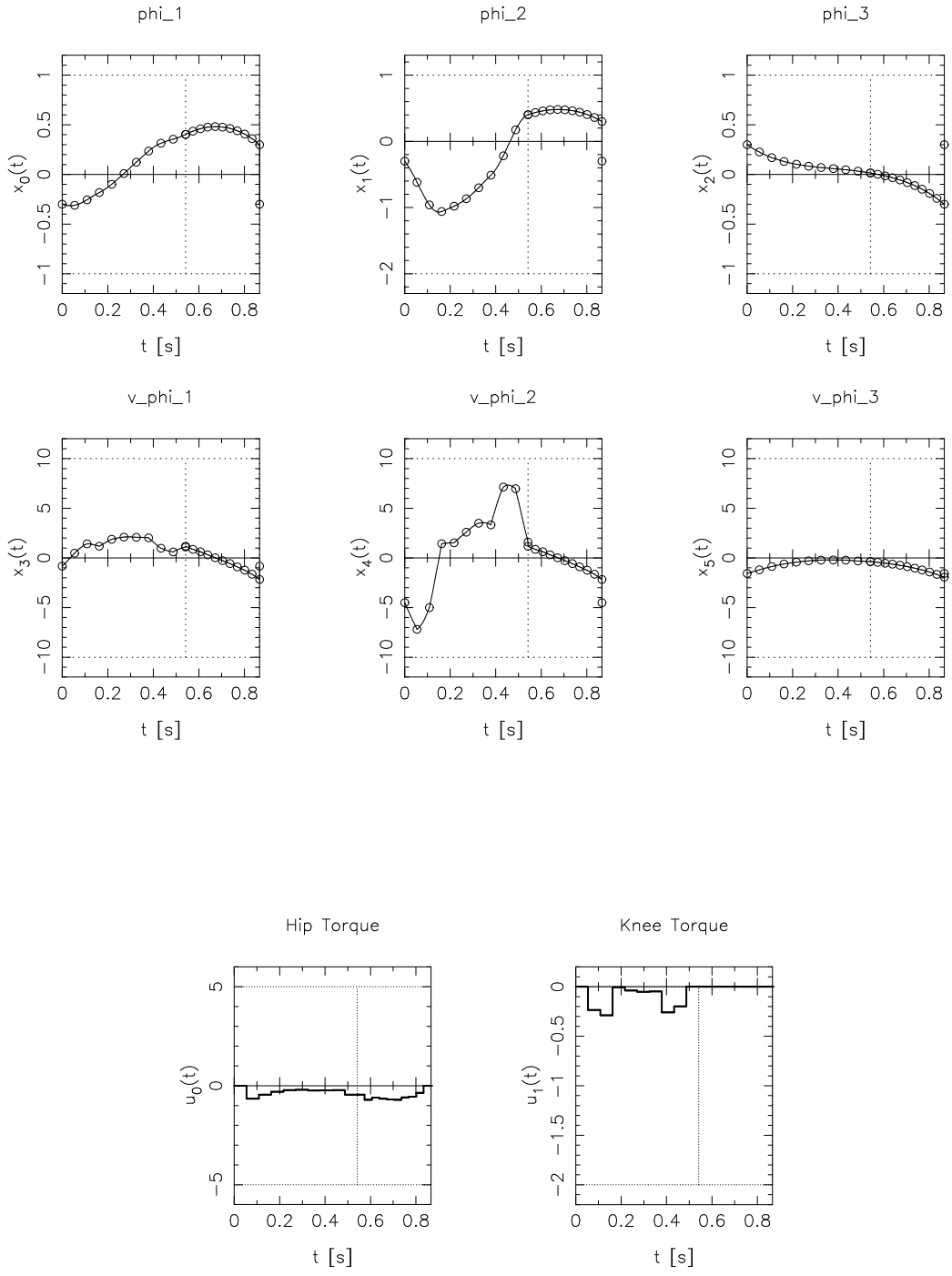


Figure 8.3: Trajectories and control variable histories of most stable solution for actuated kneed walker

$$\begin{aligned} \phi_1(t_0) &= -0.3 & \dot{\phi}_1(t_0) &= 0.8385 \\ \phi_2(t_0) &= -0.3 & \dot{\phi}_2(t_0) &= -4.5056 \\ \phi_3(t_0) &= 0.3 & \dot{\phi}_3(t_0) &= -1.5686. \end{aligned}$$

and its period is $T = 0.7832s$ with phase times $T_1 = 0.4905s$ and $T_2 = 0.2926s$.

The solution was obtained for a robot configuration with parameters values $l_T = 0.2627$, $l_S = 0.1685$, $m_T = 1.2759$, $m_S = 0.6575$, $w_S = 0.0402$, $c_T = 0.0111$, and $c_S = 0.2259$.

The robot has no variable describing the direction of travel but its step length can be computed from angular configurations and leg segments length: $\Delta_{step} = 0.127m$.

Stability computations resulted in the eigenvalues

$$\begin{aligned} \lambda_{1,2} &= (0.3091, \pm 0.475) & |\lambda_{1,2}| &= 0.5667 \\ \lambda_3 &= -0.4271 & |\lambda_3| &= 0.4271 \\ \lambda_4 &= -0.0210 & |\lambda_4| &= 0.0210 \\ \lambda_{5,6} &= 0.0 & |\lambda_{5,6}| &= 0.0. \end{aligned}$$

The maximum eigenvalue is a conjugate complex couple, and its absolute value is smaller than one. The two eigenvalues of zero come from the fact that the degrees of freedom of the robot are reduced from three to two after kneestrike (i.e. from six to four independent directions in state space). This leads to a coupling of perturbations during this second phase which is represented by a rank reduction by two of the monodromy matrix, and thus by two zero eigenvalues.

Even though the spectral radius is smaller than one, the induced matrix norms of this solution are huge – (compared e.g. to the norms computed for the hopping robot in the previous chapter):

$$\begin{aligned} \sigma_{max} &= 711.556 \\ ||C||_\infty &= 824.279 \\ ||C||_1 &= 1067.08. \end{aligned}$$

As shown in figure 8.4 they decrease with increasing matrix powers. For a cycle of 13 steps and more, the 1-, 2-, and ∞-norms are smaller than one.

Performing one-dimensional perturbations of each velocity variable and coupled perturbations of the positions which are consistent with the heelstrike manifold produces the following ranges:

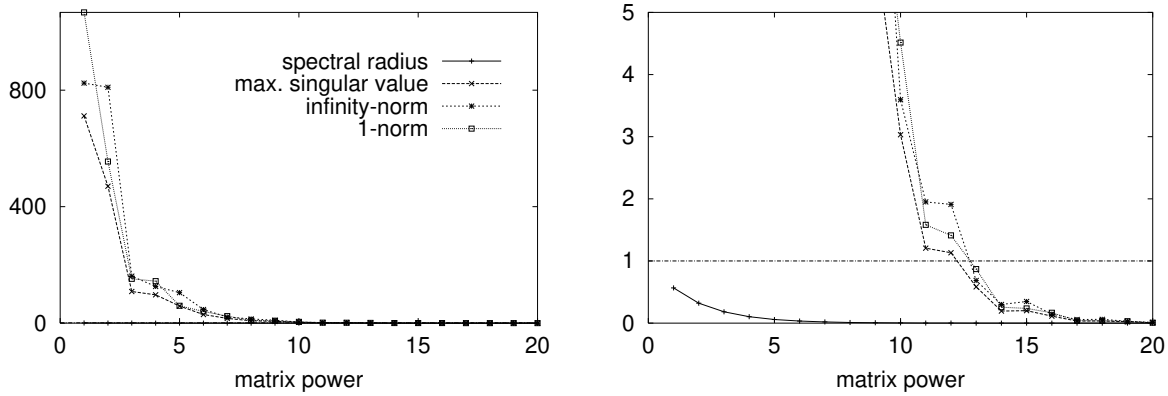


Figure 8.4: Spectral radius versus matrix norms for different matrix powers

ϕ_1 ϕ_2 ϕ_3	+0.04%	-0.02%
$\dot{\phi}_1$	+0.3%	-0.1%
$\dot{\phi}_2$	+2%	-0.4%
$\dot{\phi}_3$	+0.01%	-0.02%.

This range of attracting initial conditions could most likely be substantially increased by changing from point feet to e.g. circular feet. Further increase would probably result from making the foot a separate body and including a prescribed periodic torque at the ankle.

The maximum model parameter perturbations from which the system can still recover are:

l_T	+0.03%	-0.01%
l_S	+0.03%	-0.1%
m_T	+0.08%	-0.02%
m_S	+0.01%	-0.05%
w_S	+0.04%	-0.1%
c_T	+0.2%	-0.8%
c_S	+0.4%	-0.1%.

They are quite small but can be considered as being above the manufacturing tolerance.

We suspect that there is a correlation between the small stability margins and the large matrix norms documented above. Additionally there are nonlinear effects producing instability, like premature phase changes caused e.g. by foot scuffing in the middle of the step, that are captured neither in the eigenvalues nor in the norms. Compare figure 8.5 for switching functions and note the local minimum with small function value of the heelstrike function.

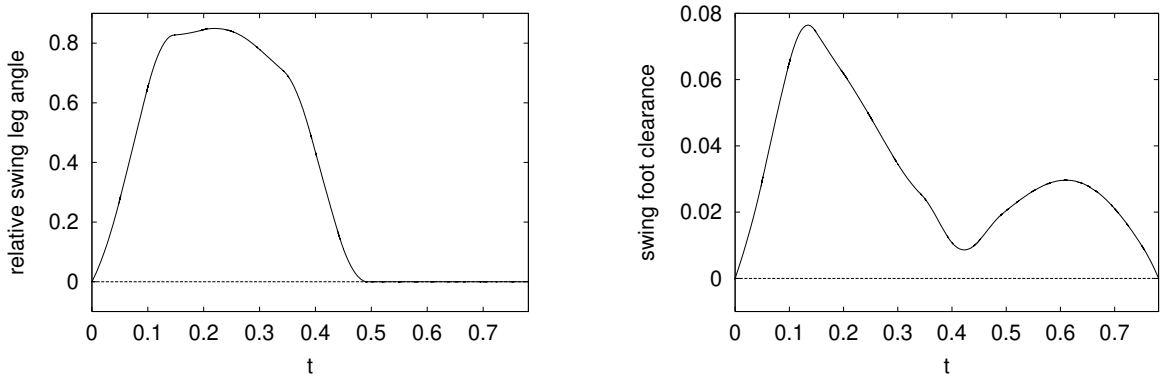


Figure 8.5: Switching functions for kneestrike (left) and heelstrike (right)

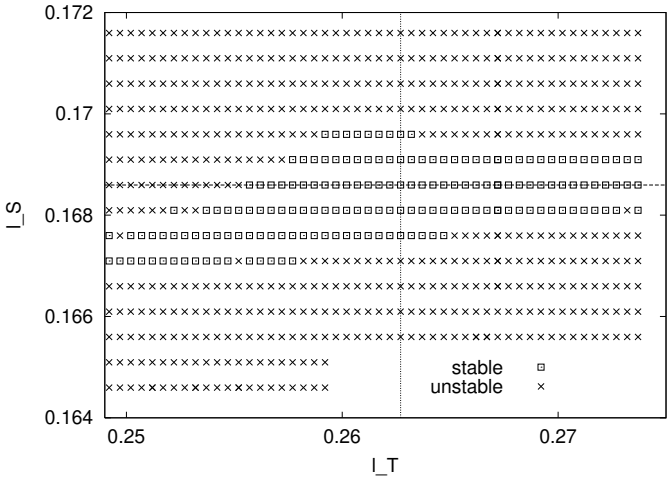


Figure 8.6: Region of stable periodic solutions in 2-dimensional parameter space (leg segment lengths l_T and l_S) in the neighborhood of most stable solution

The question we have answered so far is for which range of perturbed initial states and parameters the robot recovers under the influence of the original actuation and returns to his standard gait.

Another way to look at the question of stable areas is to check for which parameter values in the neighborhood of the solution other solutions of the periodic optimal control problem exist and are still stable. Those solutions have generally different actuator patterns and initial values. As it is impossible to visualize a seven-dimensional parameter space we present two-dimensional cuts varying only two parameters. We show a variation of masses of thigh and shank in figure 8.6, and of the respective lengths in figure 8.7, each time keeping the other five fixed. Every point in these plots represents an individual solution of the optimal control problem with joint torques minimized.

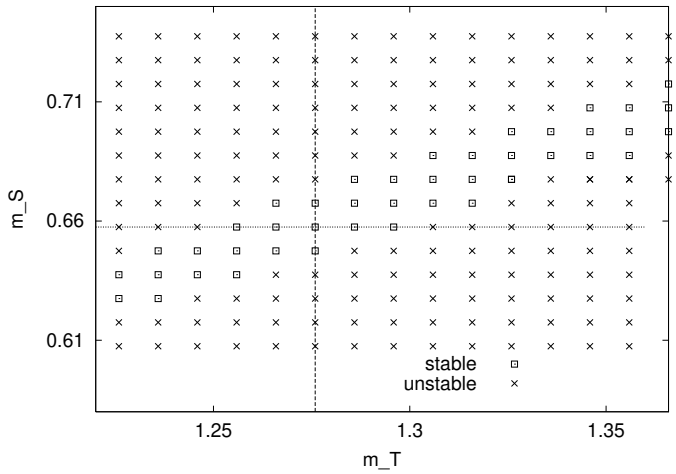


Figure 8.7: Region of stable periodic solutions in 2-dimensional parameter space (leg segment masses m_T and m_S) in the neighborhood of most stable solution

Comparison with equivalant passive-dynamic walker

We compare this most stable result for the actuated walking robot with the solution for the corresponding passive robot presented in Mombaur et al [62].

Its monodromy matrix has a spectral radius of comparable size:

$$|\lambda_{max}| = 0.6144,$$

but its nonlinear stability properties are much better.

It can sustain much more substantial perturbations of the initial values of velocities and positions than the actuated robot:

$\dot{\phi}_1$	+ 315%	- 100%
$\dot{\phi}_2$	+ 48%	- 42%
$\dot{\phi}_3$	+ 9%	- 3%
ϕ_1, ϕ_2, ϕ_3	+ 4%	- 5%

Figure 8.8 shows the motion of the robot when applying the largest possible perturbation to ϕ_1 (+ 315%) in comparison with the unperturbed motion. The self-stabilizing reaction of the system includes a very pronounced time shift. At the end the perturbed solution precedes the reference solution by about half a cycle. This reaction would be impossible for an actuated robot. Additionally, the perturbations provoke a slow oscillation about the base trajectory.

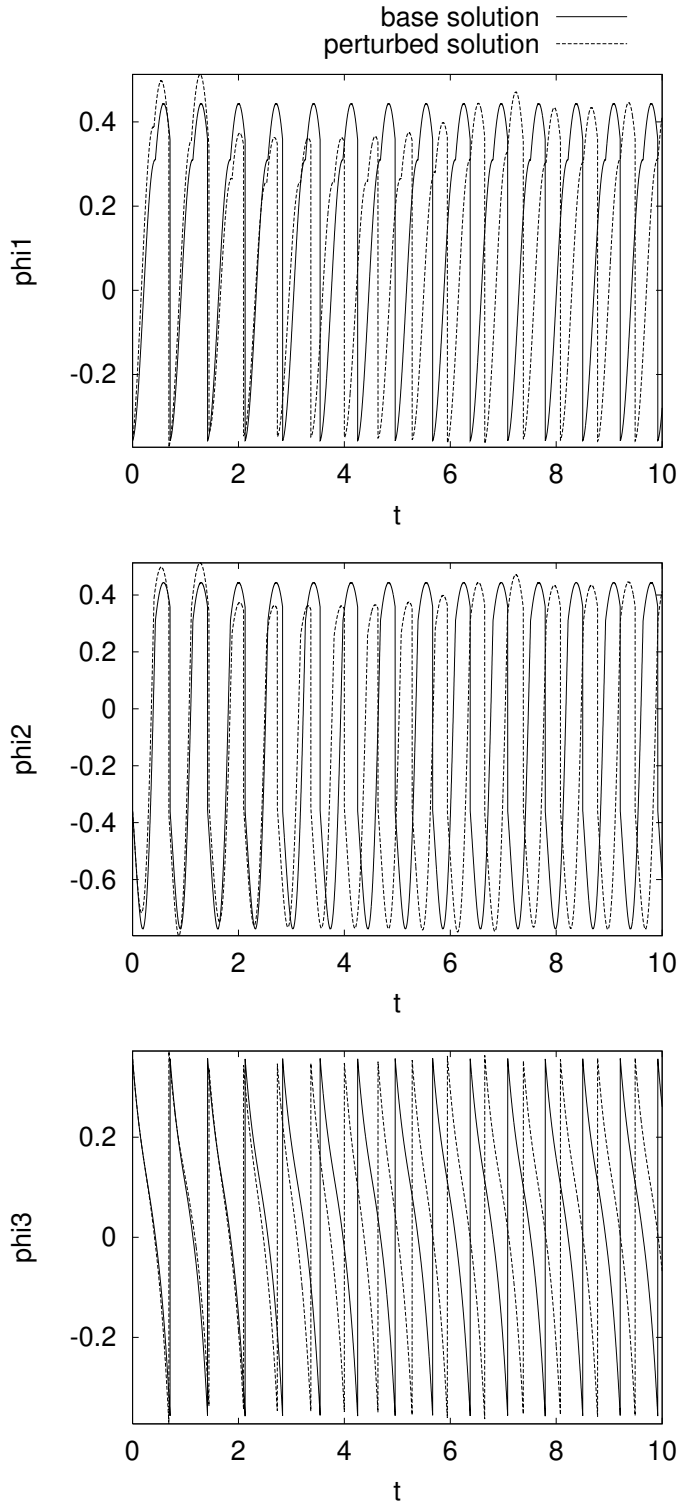


Figure 8.8: Orbital shift of perturbed trajectory versus base trajectory for corresponding passive-dynamic robot (taken from Mombaur et al. [62])

The induced matrix norms of the projected monodromy matrix are

$$\begin{aligned} \sigma_{max} &= 4.902 \\ ||C||_\infty &= 6.172 \\ ||C||_1 &= 8.099, \end{aligned}$$

much smaller than for the actuated solution.

For a complete description of this solution, we also give parameter values $l_T = 0.4017$, $l_S = 0.394$, $m_T = 2.276$, $m_S = 0.6965$, $w_S = 0.00752$, $c_T = 0.1382$, $c_S = 0.2547$, and $\alpha = 0.096$, initial point

$$x_0^T = (-0.3577, -0.3577, 0.3577, 0.2428, -2.7596, -1.3477)$$

and a period of $T = 0.7084s$ with phase times $T_1 = 0.4317s$ and $T_2 = 0.2767s$.

Result of singular value optimization

We have again studied the effect of singular value optimization of the monodromy matrix on both the maximum singular value and the maximum eigenvalue.

The result obtained from these computations is

$$\sigma_{max}(C) = 67.758$$

This is a significant reduction compared to the previous result but still a large factor of amplification for perturbations over this step.

The corresponding maximum eigenvalue is

$$|\lambda_{max}(C)| = 7.6325.$$

The solution therefore is highly unstable even judged by linear theory. Figure 8.9 illustrates the development of the spectral radius during singular value optimization with an initial deterioration and a slight improvement at the end.

The solution is characterized by the set of model parameters $l_T = 0.40$, $l_S = 0.40$, $m_T = 0.5048$, $m_S = 0.8963$, $w_S = 0.0966$, $c_T = 0.5972$, and $c_S = 0.010$ and the initial values

$$x_0^T = (-0.3, -0.3, 0.3, -0.3778, -4.0958, -1.596).$$

The period of a step is $T = 0.5814s$ with $T_1 = 0.3314s$ and $T_2 = 0.25s$ for the individual phases.

Result of Singular Value Optimization of Matrix Power

For the example of the hopping robot discussed previously, stable solutions were found by minimizing a norm of a power of the monodromy matrix. For the actuated kneed

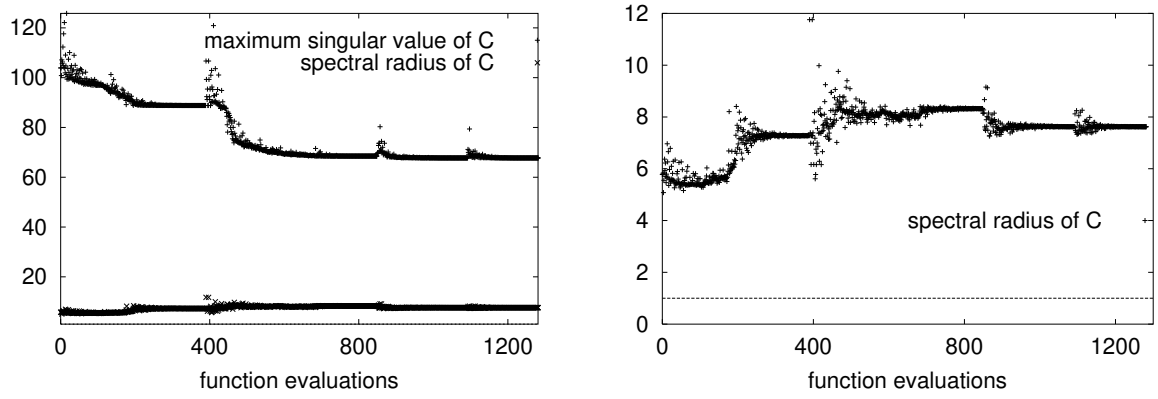


Figure 8.9: Development of spectral radius during optimization of maximum singular value of monodromy matrix

walking robot we were again successful with the same approach but had to use higher matrix powers than before.

Choosing the maximum singular value of the eighth matrix power we found an optimum of

$$\sigma_{max}(C^8) = 2.179,$$

and a corresponding spectral radius of

$$\lambda_{max}(C) = 0.906.$$

The development of the maximum eigenvalue during optimization of the maximum singular value of the eighth matrix power is shown in figure 8.11. The solution is not related to the solution found with eigenvalue optimization as it is positioned in a very distinct region in parameter space: $l_T = 0.1538$, $l_S = 0.3966$, $m_T = 0.6785$, $m_S = 1.841$, $w_S = 0.0602$, $c_T = 0.1629$, and $c_S = 0.0326$. The periodic trajectory has initial values of

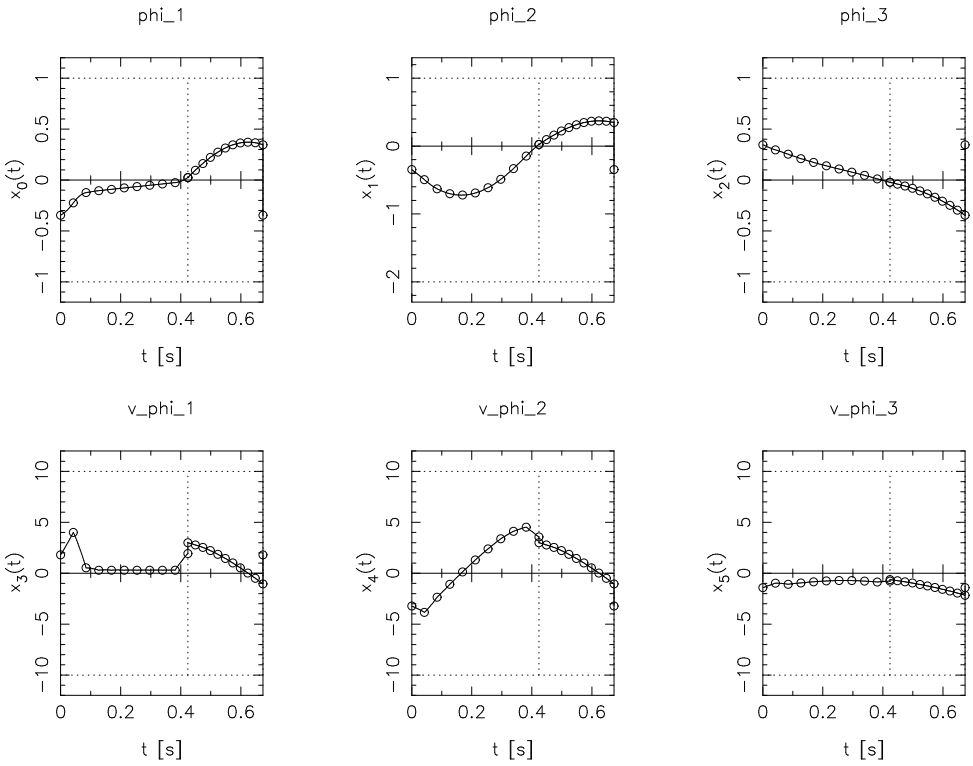
$$x_0^T = (-0.3466, -0.3466, 0.3466, 1.7952, -3.2271, -1.4128)$$

and phase times of $T_1 = 0.4234s$ and $T_2 = 0.25s$. leading to an overall cycle time of $T = 0.6734s$. Figures 8.10 gives state and control variable histories associated with this solution.

Like for the most stable solution we visualize the regions of stable solutions in the neighborhood if two out of seven parameters are varied. A variation of segments lengths l_T and l_S is shown in figure 8.12 and of segment masses m_T and m_S in figure 8.13 respectively.

The values of norms over a cycle of one step are

$$\begin{aligned} \sigma_{max}(C) &= 247.47 \\ ||C||_\infty &= 312.68 \\ ||C||_1 &= 361.68, \end{aligned}$$



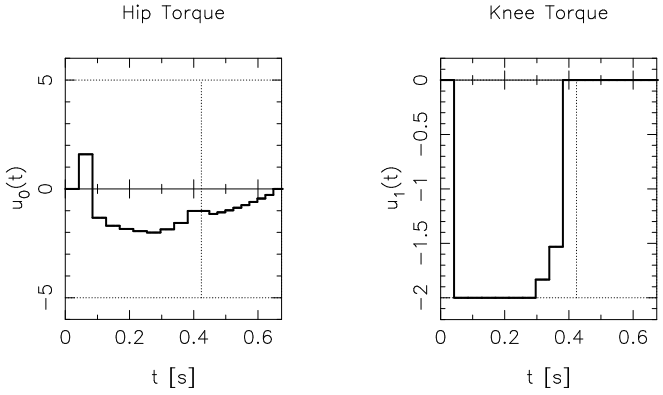


Figure 8.10: Trajectories and controls for actuated kneed walker - result of singular value optimization of C^8

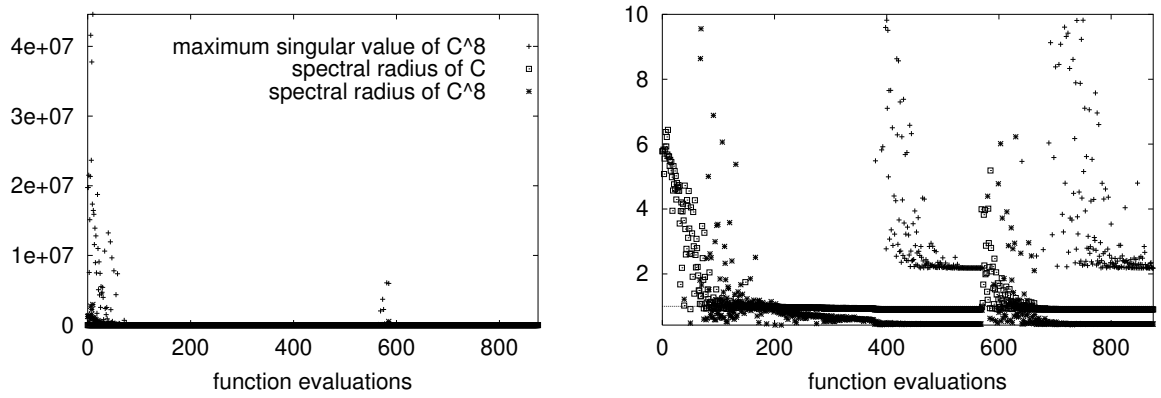


Figure 8.11: Development of spectral radius during optimization of maximum singular value of the eighth power of the monodromy matrix

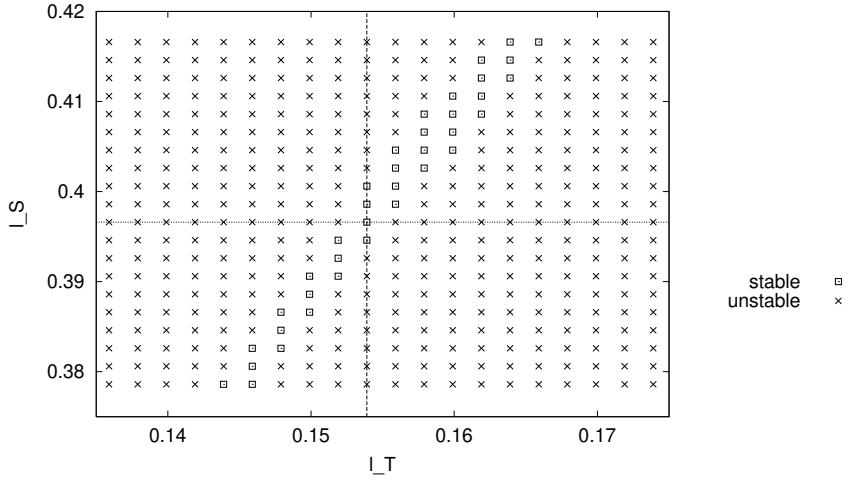


Figure 8.12: Region of stable periodic solutions in 2-dimensional parameter space (leg segment lengths l_T and l_S) in the neighborhood of solution of singular value optimization of C^8

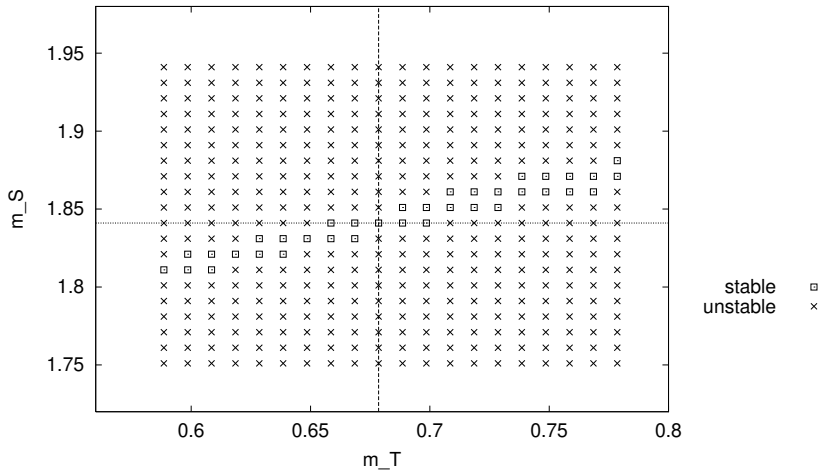


Figure 8.13: Region of stable periodic solutions in 2-dimensional parameter space (leg segment masses m_T and m_S) in the neighborhood of solution of singular value optimization of C^8

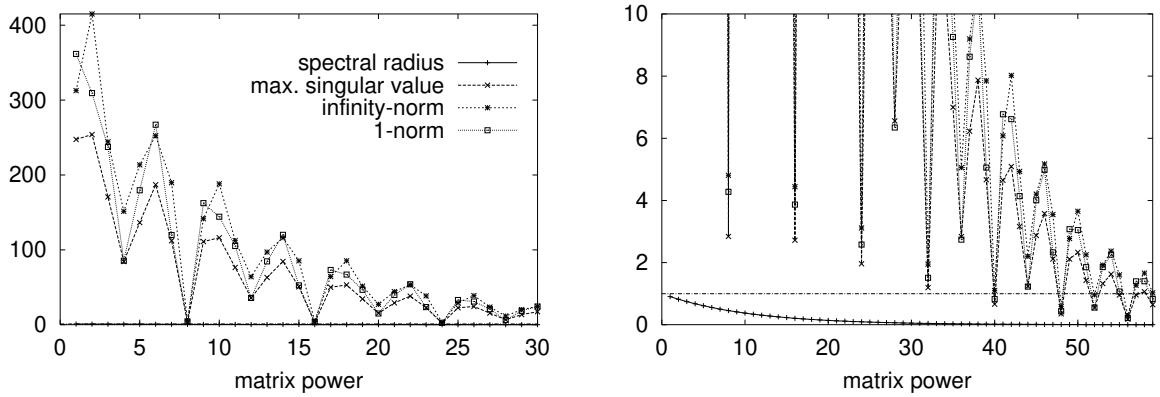


Figure 8.14: Spectral radius versus matrix norms for different matrix powers of solution of singular value optimization of C^8

still very large.

Note the behavior of norms as functions of matrix powers of this solution shown in figure 8.14. The optimization criterion causes perturbations to have the least significant amplification over multiples of eight steps.

8.3 Summary

The key results reported in this chapter are:

- We have presented stable solutions for an open-loop controlled biped robot with knees walking on flat ground in a human-like fashion actuated by torques at hip and knees. To our knowledge this is the first robot of this kind.
- Judged by linear theory the best solution found is very stable with a maximum eigenvalue safely below one. Its stability margins describing possible perturbations of initial values and parameters are not very large. The regions of stable periodic solutions in 2-dimensional parameter space are more extended and have band shape.
- Comparison with an equivalent passive biped walker shows that the passive system has considerably larger stability margins even though its monodromy matrix has roughly the same spectral radius. During recovery from perturbations the passive system encounters an enormous shift along its trajectory. We have seen however in the previous chapter that small stability margins are not a standard feature of actuated open-loop controlled robots.
- With respect to matrix norm optimization the observations reported in the previous chapter could be confirmed:
  1. There are again no solutions for which perturbations decay over one cycle measured in the 1-, 2, or ∞-norm.
  2. Optimization of a norm of the monodromy matrix itself results in an unstable solution.
  3. Optimization of a norm of a higher matrix power delivers stable solutions. There is no general rule for the choice of the matrix exponent. In this case a higher power than previously had to be used.
- While asking for a norm smaller than one is too strict and not necessary, the norm also should not be too large. We have observed a correlation between small stability margins and extremely large matrix norms (> 100), although we are aware that this is not the only reason for destabilization. There are also nonlinear effects, like e.g. discontinuous changes of the switching structure.

Chapter 9

Three-dimensional Passive-dynamic Walking Robot – The Tinkertoy

The Tinkertoy robot is a passive three-dimensional walker with two straight legs that moves on an inclined slope without any actuator help. To our knowledge it is the first three-dimensional dynamically stable robot that has no statically stable standing position.

The physical robot has been built by Coleman [18] experimenting with the Tinkertoy® construction set. Before our cooperation started, the stable behavior of the real robot could not be verified theoretically; all simulations of the model had been unstable (Coleman& Ruina [20]). At this point it was not clear if a statically unstable rigid body model could be passively stable in three dimensions or if the dynamic stability of the real robot was due to properties not captured in the mathematical model, e. g. the link elasticity. The goal of our computations was not exactly mimic the quantities of the physical robot, but to answer this more general question and find stable configurations more or less related to the real robot.

The mathematical model has been established by Coleman using the MATLAB® software package. Corrections and modifications have been done in joint work. Kinematic relations for the motion of robots with different foot forms have been established on the basis of Goyal [34]. For use with our optimization software we have transferred the model equations to C++ and included it in our model library.

In this thesis, we give results for three different model versions with disk feet, toroidal feet and point feet, where the third is a special case of the first as well as the second with radii equal to zero. This is the first publication of stable solutions for the point feet and toroidal feet versions whereas the disk foot results are an improvement of the stable solutions published in Mombaur et al. [63] and Coleman and al. [19].

According to the classification of section 1.4 the Tinkertoy robot is non-conservative due to ground collision and non-holonomic (disk and toroidal feet) or piecewise holonomic but overall non-holonomic (point feet). Both aspects can contribute to the existence of stable solutions.

9.1 Robot Model

The Tinkertoy robot consists of two legs, i.e. a pair of symmetric rigid bodies, connected by a hinge. Always only one foot is in rolling, non-sliding contact with the ground. The foot switching is assumed to be instantaneous and collisional such that there is no double support phase. Figure 9.1 contains an animation of the Tinkertoy with disk shaped feet. The robot model with its parameters and configuration variables is further specified in figure 9.2. We model only one step and not a full cycle consisting of two steps. This procedure eliminates unsymmetric gaits as well as oblique directions of descent. The heelstrike collision also includes the leg shift from left to right and vice versa to make the application of periodicity constraints after one step possible.
The system has got four DOFs. For the description, we use the following angle coordinates

$$q^T = (\phi, \psi, \theta_{st}, \theta_{sw})^T \tag{9.1}$$

where, speaking in aeronautical terms, ϕ is the robot's heading angle, ψ is its rolling angle, θ_{st} the pitch angle of the stance leg, and θ_{sw} the relative pitch orientation of the swing leg. With the corresponding rates this results in an eight-dimensional state space.
The inertia matrix of a rigid body in three dimensions has to satisfy the two properties:

- all eigenvalues are positive
- eigenvalues satisfy the triangle inequality $\lambda_i + \lambda_j > \lambda_k$, $i \neq j \neq k$.

To guarantee a fulfillment of these properties during variation of the inertia matrix, Coleman and Ruina have developed a re-parameterization (see Coleman [18]). The principal moments of inertia can be rewritten using the parameters d_1, d_2, d_3 (describing the dynamically equivalent arrangement of six masses $m_d = \frac{m}{6}$ in the distances d_i from the center):

$$I_1 = (d_2^2 + d_3^2) \tag{9.2}$$
$$I_2 = (d_1^2 + d_3^2) \tag{9.3}$$
$$I_3 = (d_1^2 + d_2^2). \tag{9.4}$$

A general inertia matrix is generated from the principal axis inertia by rotations characterized by three additional parameters, the angles ρ, β, γ.
The 14/15 model parameters of the tinkertoy robot are the six inertia parameters d_1, d_2, d_3, ρ, β, γ, the leg mass m, the slope angle α, the total leg length l, the leg center of mass location d_x, d_y, d_z in local leg coordinates the hip spacing w, the foot radius r_1, and for toroidal feet additionally a second (perpendicular) foot radius r_2. Please note that all parameters are dimensionless: all lengths are measured relative to the total leg length and all inertia matrix entries are relative to ml^2 (and the d_i therefore relative to $\sqrt{ml}$), and the gravity constant is set to 1.
The equations of motion of the Tinkertoy robot are too complex to be given in explicit form. The four second order equations of motion are derived by angular momentum balances of

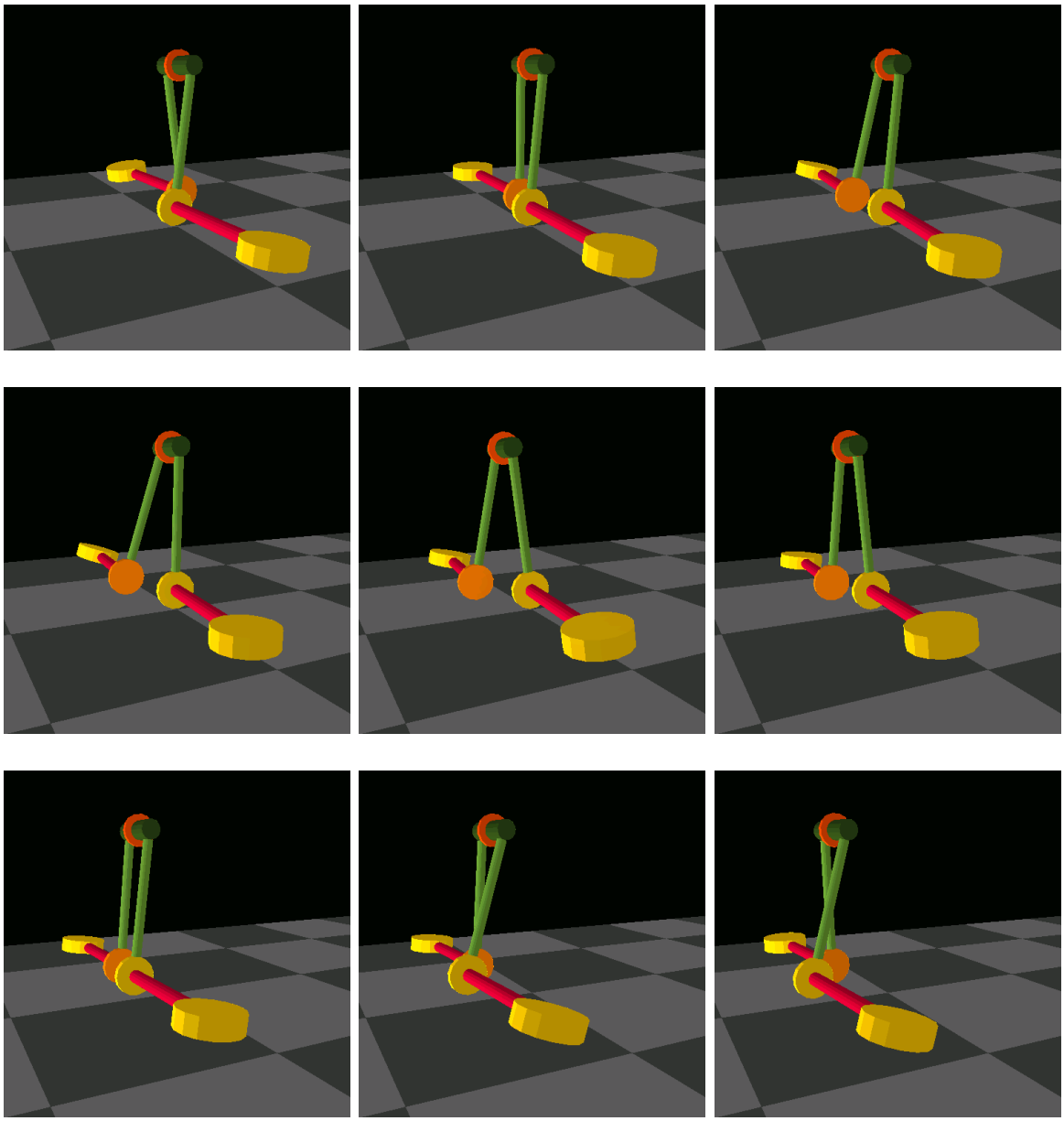
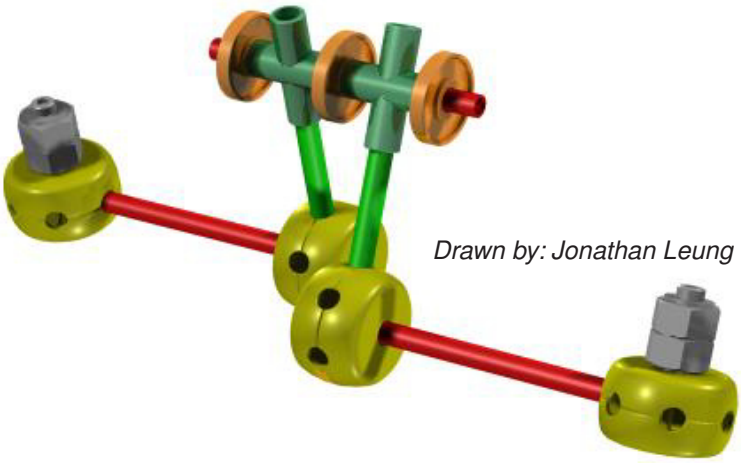


Figure 9.1: Stable periodic gait of Tinkertoy robot with disk feet



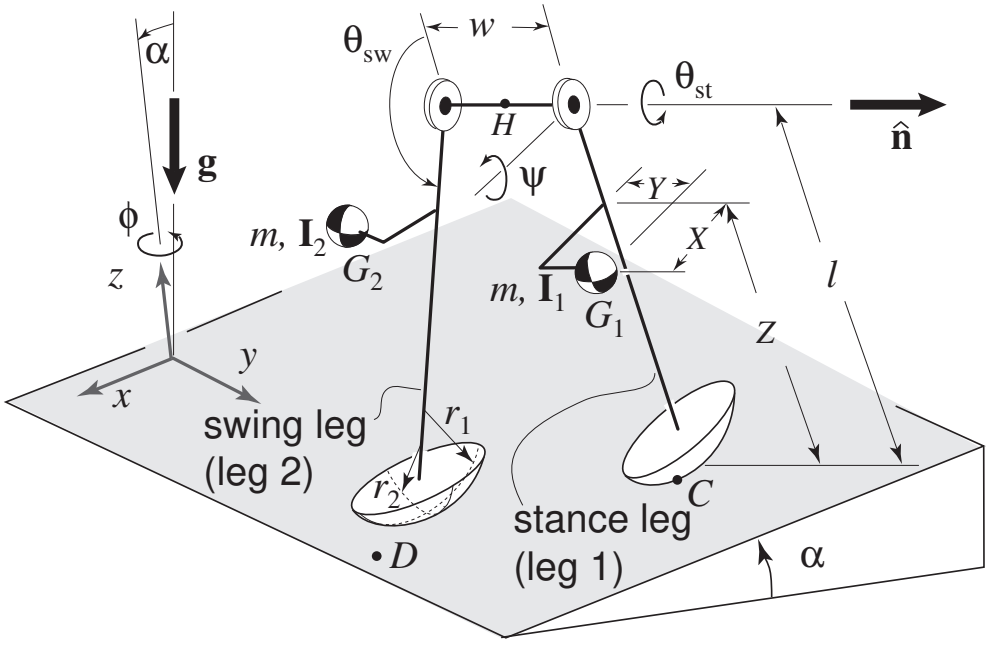


Figure 9.2: The real Tinkertoy robot and its model (drawings by Coleman and Leung)

- the whole robot about contact point C of stance leg

$$\sum_{i=1}^{2} r_{C,i} \times m_i a_i + \sum_{i=1}^{2} (\Theta_i \dot{\omega}_i + \omega_i \times \Theta_i \cdot \omega_i) = \sum_{i=1}^{2} r_{C,i} \times m_i g \tag{9.5}$$

  where $r_{C,i} = r_i - r_C$ is the position of the center of mass of body i relative to the contact point C, $a_i = \ddot{r}_i$ is the absolute acceleration of body i, ω_i its absolute angular velocity, Θ_i its inertia matrix in local coordinates, and m_i its mass (with all masses being equal: $m_1 = m_2 = m$).

- the swing leg about the hip joint hinge axis n_{hip} (index 1 denotes the stance leg and 2 the swing leg):

$$n_{hip} \cdot (r_{hip,2} \times m_2 a_2 + \Theta_2 \dot{\omega}_2 + \omega_2 \times \Theta_2 \cdot \omega_2) = n_{hip} \cdot (r_{hip,2} \times m_2 g)\,. \tag{9.6}$$

Ground collision of the swing foot occurs when its lowest point reaches zero altitude. As both feet have the same rotational symmetric shape and the legs rotate in parallel planes this is for all possible foot shapes equivalent with the postulation that the centers (in terms of r_1) of both feet are at the same level:

$$s(x,p) = (l - r_1)(\cos(\theta_{st} + \theta_{sw}) + \cos\theta_{st}) \cos\psi - w \sin\psi = 0. \tag{9.7}$$

As an additional condition, the vertical velocity of the lowest swing foot point (or alternatively the derivative of the above equation) has to be negative:

$$\begin{aligned} c(x,p) \;=\; & (l - r_1)(\cos(\theta_{st} + \theta_{sw}) + \cos\theta_{st})(-\sin\psi\dot{\psi}) + l(-\sin(\theta_{st} + \theta_{sw})(\dot{\theta}_{st} + \dot{\theta}_{sw}) - \\ & \sin\theta_{st}\dot{\theta}_{st}) \cos\psi + w \cos\psi\dot{\psi} < 0 \end{aligned} \tag{9.8}$$

At collision, position variables are of course continuous in physical space, but velocities undergo discontinuities. Velocities after heelstrike (and after foot switching) are computed with the following relations:

- conservation of angular momentum of the whole system about the new contact point D:

$$\sum_{i=1}^{2} (r_{D,i} \times m_i v_i + \Theta_i \omega_i) = const. \tag{9.9}$$

  where $r_{D,i} = r_i - r_D$ and v_i is the absolute velocity of body i.

- conservation of angular momentum of the new swing leg $sw+$ (index 1 before and index 2 after collision) about the hinge axis:

$$n_{hip} \cdot (r_{hip,sw+} \times m_{sw+} v_{sw+} + \Theta_{sw+} \omega_{sw+}) = const. \tag{9.10}$$

During the whole swing phase, foot clearance of the swing foot is enforced by:

$$l(\cos(\theta_{sw} + \theta_{st}) + \cos\theta_{st}) \cos\psi - w \sin\psi > 0. \tag{9.11}$$

We demand periodicity of all eight state variables over one step, including the leg shift:

$$\begin{pmatrix} \phi \\ \psi \\ \theta_{st} \\ \theta_{sw} \\ \dot{\phi} \\ \dot{\psi} \\ \dot{\theta}_{st} \\ \dot{\theta}_{sw} \end{pmatrix} (T) = \begin{pmatrix} \phi \\ \psi \\ \theta_{st} \\ \theta_{sw} \\ \dot{\phi} \\ \dot{\psi} \\ \dot{\theta}_{st} \\ \dot{\theta}_{sw} \end{pmatrix} (0) \tag{9.12}$$

with free cycle time T.

9.2 Results of Stability Optimization

We present stable solutions for Tinkertoy robots with disk feet, toroidal feet and point feet. In all three cases we used the spectral radius as objective function in the outer stability optimization loop. Box constraints have been imposed on the parameters in order to keep them within reasonable ranges and to avoid non-physical results, like negative dimensions.

In the inner loop periodic optimal control problem – with zero controls – we have minimized the duration of steps for all solutions presented here. We have used ten multiple shooting intervals for the swing phase. The foot clearance condition has been imposed on all interior points. State variables must satisfy box constraints. To guarantee a minimum steplength and avoid the degenerate case of zero step length a condition on $\theta_{st}(0)$ has been formulated.

9.2.1 Disk Feet

As a start, results for the model version with disk feet will be presented and explained. For this model we have already published stable solutions in Mombaur et al. [63] and Coleman and al. [19]. During the research for this thesis we were able to further improve the results by means of the two-level optimization procedure and perform a detailed analysis of the solutions.

Result of eigenvalue optimization

The most stable solution in terms of eigenvalues for the Tinkertoy robot with disk feet is characterized by a monodromy matrix with spectral radius 0.7579, safely below one. It represents the overall optimum of eigenvalue optimization.

The resulting robot configuration has the model parameters $d_1 = 0.1442$, $d_2 = 0.393$, $d_3 = 0.2925$, $\rho = -0.0138$, $\beta = -0.2688$, $\gamma = -8.2519E-03$, $m = 1.0$, $\alpha = 0.0757$,

$d_x = 0.0029$, $d_y = 0.7903$, $d_z = 0.4064$, $l = 1.0$, $w = 0.3234$, and $r_1 = 0.032$, all without dimension. Note that the robot is statically unstable since $d_z > r_1$. The resulting radius of the disk foot in the optimum is surprisingly small.

The initial values of this most stable solution are

$$\begin{aligned} \phi(0) &= 0.0969 & \dot{\phi}(0) &= -0.0863 \\ \psi(0) &= -0.0132 & \dot{\psi}(0) &= -0.0275 \\ \theta_{st}(0) &= -0.196 & \dot{\theta}_{st}(0) &= 0.410 \\ \theta_{sw}(0) &= 3.510 & \dot{\theta}_{sw}(0) &= -0.327 \end{aligned}$$

Figure 9.3 shows the corresponding trajectory for all eight state variables. It pictures one full step of the robot consisting of the swing phase and the final discontinuity at heelstrike. The heelstrike transition phase also includes a leg shift which explains why the plots show discontinuities not only of the velocities but also of the position variables. All eight state variables satisfy periodicity conditions. The duration of one step is $T = 1.358$. In figure 9.4 we show swing foot clearance over one step which is the switching function for heelstrike detection.

The eight eigenvalues of the Jacobian of the Poincaré map associated with this solution are

$$\begin{aligned} \lambda_1 &= 1.0 \\ \lambda_{2,3} &= (-0.0076, \pm 0.7575) \\ \lambda_{4,5} &= (-0.6962, \pm 0.2996) \\ \lambda_{6,7} &= (-0.7048, \pm 0.27658) \\ \lambda_8 &= -0.3112 \end{aligned}$$

and in terms of absolute values

$$\begin{aligned} |\lambda_1| &= 1.0 \\ |\lambda_{2,3}| &= 0.7575 \\ |\lambda_{4,5}| &= 0.7579 \\ |\lambda_{6,7}| &= 0.7571 \\ |\lambda_8| &= 0.3112 \end{aligned}$$

The eigenvalue of one is caused by the passivity of the Tinkertoy and, as we have explained in section 4.4, is not relevant for stability. $|\lambda_{4,5}|$ represents the spectral radius of the monodromy matrix. The fact that six out of seven parameters are equal within the tolerance of convergence is an indication that the optimization has produced not only a local minimum but also a point that is very good from a global point of view.

The induced matrix norms of the projected matrix are

$$\sigma_{max}(C) = 4.284$$

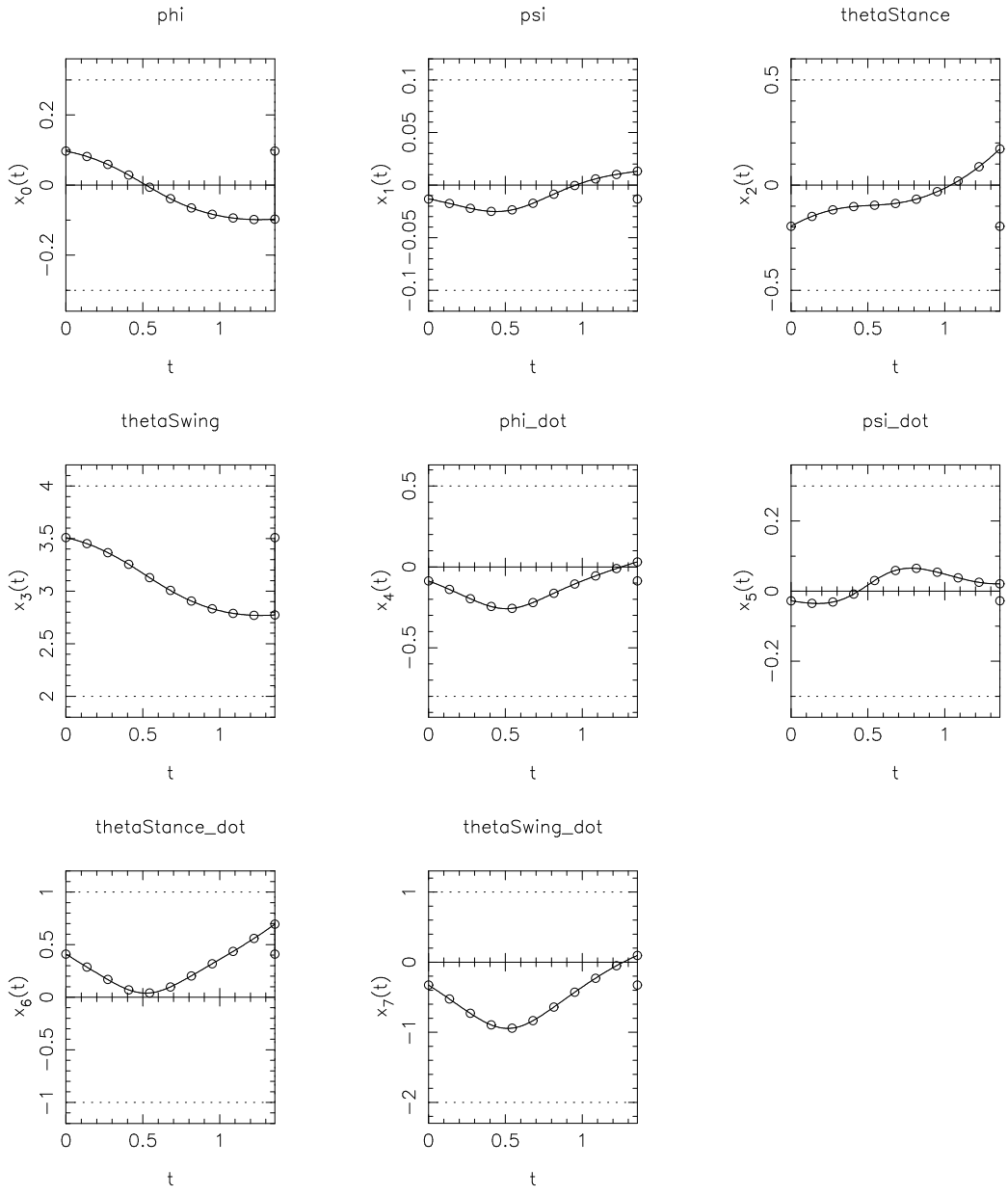


Figure 9.3: Most stable trajectory of Tinkertoy with disk feet

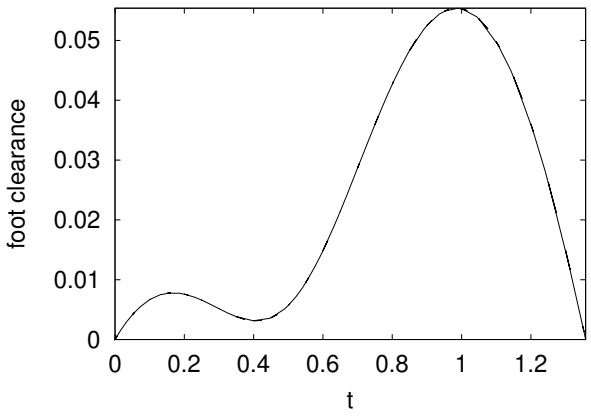


Figure 9.4: Swing foot clearance during one step

$$\begin{aligned} ||C||_\infty &= 6.260 \\ ||C||_1 &= 7.892 \end{aligned}$$

Measured in these norms perturbations do therefore not contract but are locally amplified in this particular step. Figure illustrates the size of matrix norms for increasing matrix powers, i.e. for multiple steps. For this specific solution a contraction of perturbations only occurs over a cycle of 18 steps or more than 22 steps.

The study of stability margins has however proven again that a contraction of norms over one step is not necessary for stable behavior of the discontinuous system. We were able to apply one-dimensional perturbations of considerable size to the initial values of all position and velocity variables, from which the system would still recover:

ϕ	+39%	-43%
ψ	+56%	-56%
θ_{st}	+5%	-4%
θ_{sw}	+0.6%	-1%
$\dot{\phi}$	+8%	-6%
$\dot{\psi}$	+19%	-14%
$\dot{\theta}_{st}$	+2%	-2%
$\dot{\theta}_{sw}$	+5%	-5%

The reason for failure if larger perturbations are applied is typically foot scuffing in the middle of the step (compare foot clearance function in figure 9.4).

Figure 9.6 illustrates the decay of the oscillation introduced by a perturbation of +39% applied to the initial value of ϕ over a longer interval. In figure 9.7 we compare the trajectories of this perturbed solution to the corresponding base solution over a period of a little bit more than ten steps. The system recovers from this perturbation while only performing a small orbital shift.

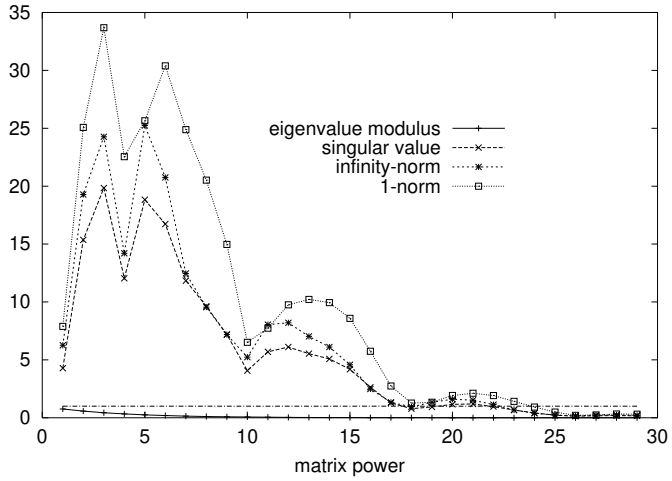


Figure 9.5: Matrix norms and spectral radius for different matrix powers of monodromy matrix over different numbers of steps

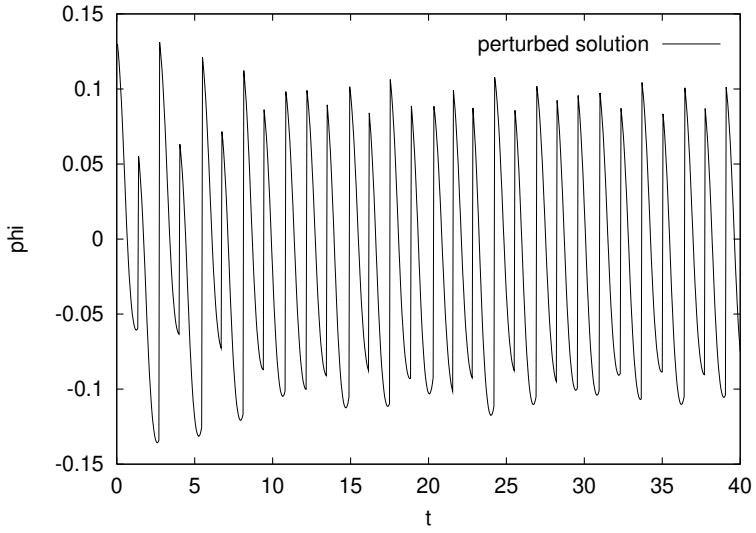


Figure 9.6: Decay of the oscillation caused by a perturbation

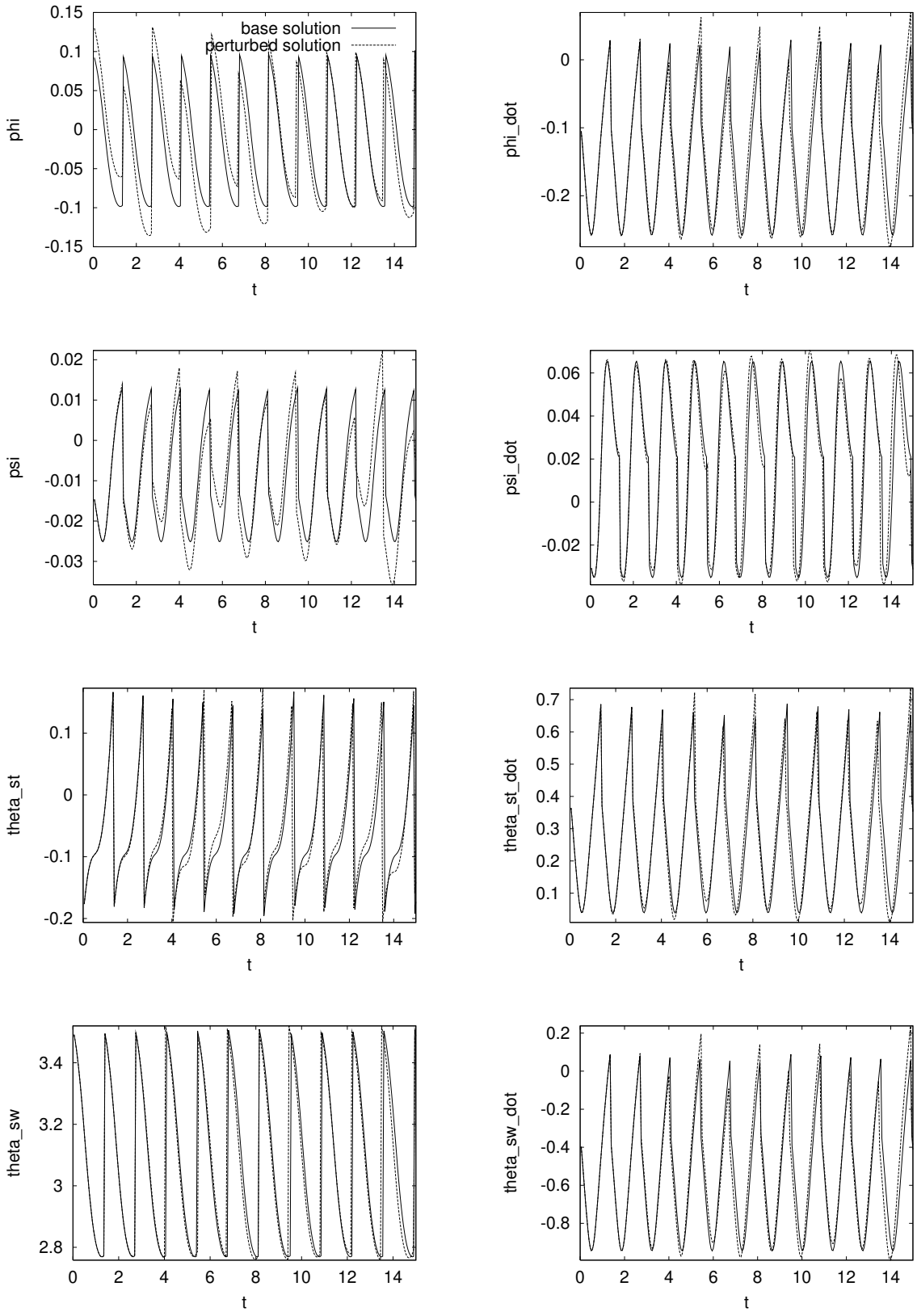


Figure 9.7: Most stable periodic solution of Tinkertoy with disk feet with and without perturbation of ϕ

The parameter values could likewise be perturbed within large ranges:

d_1	+40%	-240%
d_2	+64%	-264%
d_3	+12%	+14%
ρ	+2300%	-2600%
β	+1300%	-1500%
γ	$+\infty$	$-\infty$
α	+17%	-30%
d_x	+130%	-140%
d_y	+8%	-8%
d_z	+7%	-5%
w	+12%	-7%
r_1	+72%	-75%

Another stable solution

We present a second stable solution with a spectral radius of 0.809, slightly larger than for solution 1. It has been found as intermediate convergence point of eigenvalue optimization before a restart of the polytope algorithm.

This solution is interesting for different reasons:

- it shows that stable solutions exist in different regions of parameter space,
- although the eigenvalue indicates weaker stability, its stability margins are larger than for solution 1,
- for some purposes it might be useful to have a stable robot with a more pronounced disk foot.

The model parameters of this robot are $d_1 = 0.0074$, $d_2 = 0.8805$, $d_3 = 0.021$, $\rho = 0.0456$, $\beta = -0.301$, $\gamma = 0.0049$, $m = 1.0$, $\alpha = 0.077$, $d_x = -4.7e-5$, $d_y = 0.7024$, $d_z = 0.1856$, $l = 1.0$, $w = 0.3579$, and $r_1 = 0.1185$. The most significant differences to the first solution lie in the mass distribution, the foot radius and the vertical c. o. m. position.

The initial value of the corresponding trajectory which looks very similar to the previous one are

$$\begin{aligned} \phi(0) &= 0.1044 & \dot{\phi}(0) &= -0.1233 \\ \psi(0) &= -0.0102 & \dot{\psi}(0) &= -0.0218 \\ \theta_{st}(0) &= -0.1729 & \dot{\theta}_{st}(0) &= 0.4727 \\ \theta_{sw}(0) &= 3.462 & \dot{\theta}_{sw}(0) &= -0.3746 \end{aligned}$$

The cycle time is $T = 1.130s$.

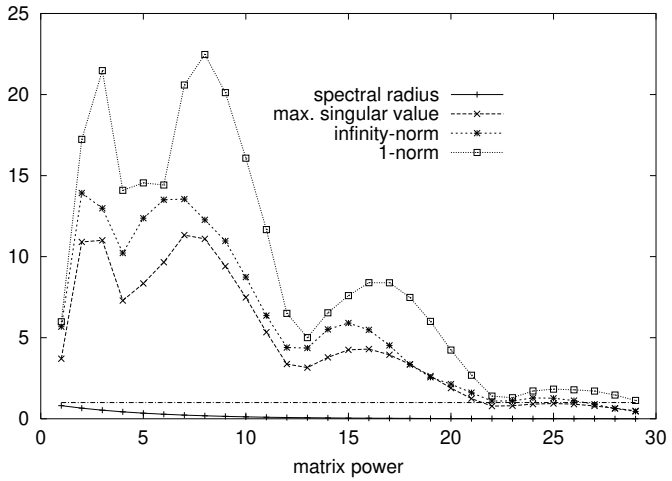


Figure 9.8: Matrix norms and spectral radius for different matrix powers of monodromy matrix over different numbers of steps

It is characterized by the eigenvalues

$$\begin{aligned} \lambda_1 &= 1.0 & |\lambda_1| &= 1.0 \\ \lambda_{2,3} &= (0.2432, \pm 0.6403) & |\lambda_{2,3}| &= 0.685 \\ \lambda_{4,5} &= (-0.7724, \pm 0.2395) & |\lambda_{4,5}| &= 0.8087 \\ \lambda_{6,7} &= (-0.7657, \pm 0.2614) & |\lambda_{6,7}| &= 0.8091 \\ \lambda_8 &= -0.2752 & |\lambda_8| &= 0.2752. \end{aligned}$$

Again the 1-, 2-, and ∞-norm are not contractive for this one-step cycle

$$\begin{aligned} \sigma_{max}(C) &= 3.705 \\ ||C||_\infty &= 5.682 \\ ||C||_1 &= 5.986. \end{aligned}$$

The maximum singular value is contractive for a cycle of more than 22 steps (see figure 9.8).

The stability margins of this solution are larger for the previous one in most components of the state variable vector:

ϕ	+100%	-102%
ψ	+136%	-95%
θ_{st}	+15%	-14%
θ_{sw}	+1%	-1%
$\dot{\phi}$	+9%	-9%
$\dot{\psi}$	+32%	-32%
$\dot{\theta}_{st}$	+6%	-5%
$\dot{\theta}_{sw}$	+7%	-9%

This solution also allows for larger perturbations of the parameter values under which the robot can still recover:

d_1	+2000%	-2000%
d_2	+27%	-19%
d_3	+230%	-430%
ρ	+11%	-9%
β	+3%	-0.5%
γ	$+\infty$	$-\infty$
α	+29%	-62%
d_x	+ 10000%	- 10000%
d_y	+10%	-23%
d_z	+27%	-36%
w	+22%	-8%
r_1	+26%	-30%

9.2.2 Toroidal Feet

The model version with toroidal feet is the most general of the three as it includes the other two as special cases. A model with feet of non-degenerate toroidal shape also constitutes the best approximation of the real Tinkertoy robot. We did not aim at imitating the quantitities of the real robot in our computations but rather at finding the solution with the best stability properties.

Result of eigenvalue optimization

Eigenvalue optimization for the Tinkertoy with toroidal feet produced an optimal value of 0.7571. Treating the most general case, it is lower than the disk feet solution, as one could expect. But obviously there is not a big difference which can be explained by the parameter values at the solution: $d_1 = 0.0483$, $d_2 = 0.420$, $d_3 = 0.3191$, $\rho = 0.0545$, $\beta = -0.3072$, $\gamma = -0.0097$, $m = 1.0$, $\alpha = 0.075$, $d_x = 0.0027$, $d_y = 0.7949$, $d_z = 0.4396$, $l = 1.0$, $w = 0.3339$, $r_1 = 0.01827$ $r_2 = 0.0053$. The foot radii are surprisingly small, especially the second one. The optimal toroidal foot is thus close to a tiny disk - and not too far from a point foot.

The fixed point of this periodic solution is

$$x_0^T = (0.093, -0.0124, -0.1962, 3.511, -0.0835, -0.0291, 0.402, -0.3295)$$

and its cycle time $T = 1.3536s$.

The full set of eigenvalues of Jacobian of the Poincaré map is

$$\begin{aligned}
\lambda_1 &= 1.0 & |\lambda_1| &= 1.0 \\
\lambda_{2,3} &= (0.0161, \pm 0.7569) & |\lambda_{2,3}| &= 0.7571 \\
\lambda_{4,5} &= (-0.6991, \pm 0.291) & |\lambda_{4,5}| &= 0.7572 \\
\lambda_{6,7} &= (-0.7044, \pm 0.2777) & |\lambda_{6,7}| &= 0.7572 \\
\lambda_8 &= -0.308 & |\lambda_8| &= 0.308.
\end{aligned}$$

The induced matrix norms of the projected map are

$$\begin{aligned}
\sigma_{max} &= 4.324 \\
||C||_\infty &= 6.342 \\
||C||_1 &= 7.835,
\end{aligned}$$

being even slightly larger than those of the most stable solution for a disk foot walker.

Applying one-dimensional perturbations to the initial values of each state variable we determined the following stability margins:

ϕ	+55%	-66%
ψ	+65%	-62%
θ_{st}	+5%	-4%
θ_{sw}	+0.5%	-1%
$\dot{\phi}$	+9%	-6%
$\dot{\psi}$	+18%	-14%
$\dot{\theta}_{st}$	+2%	-2%
$\dot{\theta}_{sw}$	+6%	-5%.

Figure 9.9 illustrates the reaction of a robot to a perturbation of the initial value of $\dot{\phi}$ (9%). Note the significant orbital shift and the large oscillations about the base trajectory in some components of the state variable vector.

Model parameters could also be perturbed by large percentages:

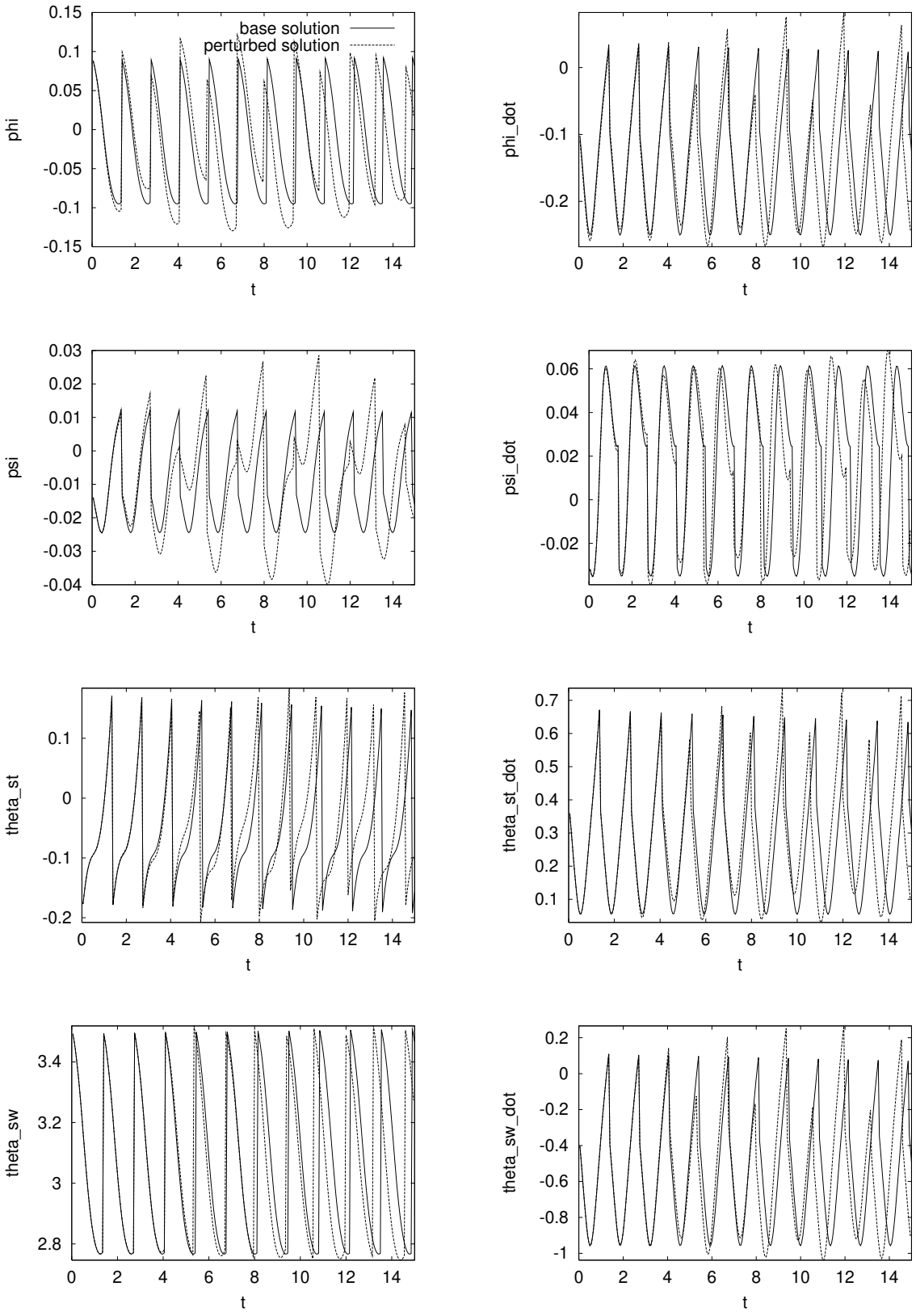


Figure 9.9: Stable periodic solution for Tinkertoy with toroidal feet with and without perturbation of $\dot{\phi}$

d_1	+220%	-420%
d_2	+62%	-262%
d_3	+10%	-12%
ρ	+490%	-540%
β	+120%	-310%
γ	$+\infty$	$-\infty$
α	18%	-30%
d_x	+130%	-170%
d_y	7%	-7%
d_z	8%	-5%
w	+12%	-7%
r	+130%	-100%
r_2	+700%	-100%.

Results of norm optimization

In the two previous chapters the effects of norms as optimization criterion on the spectral radius have been discussed extensively. Computations for the Tinkertoy confirm the results presented before and we will therefore skip the discussion.

Here we want to focus on the question if at least for this passive system solutions with contracting norms of the monodromy matrix can be found. We have studied the question for this most general case of the Tinkertoy with toroidal feet. The answer is again negative.

Singular value optimization resulted in an optimum of

$$\sigma_{max}(C) = 2.854.$$

Minimizing the ∞-norm of the monodromy matrix lead to a final value of

$$||C||_\infty = 4.354.$$

Of course this is not a general rule for passive systems, but a specific result for the Tinkertoy. We can only conclude that asking for a 1-, 2-, or ∞-norm below one seems to be a demand difficult to satisfy.

9.2.3 Point Feet

The existence of stable solutions for a walker with point feet is probably the most astonishing result reported in this chapter. It is a special case of the latter two with $r_1 = r_2 = 0$.

Result of eigenvalue optimization

The most stable solution for the Tinkertoy robot with disk feet has a monodromy matrix with spectral radius 0.7958. The eight eigenvalues are

$$\begin{aligned}
\lambda_1 &= 1.0 & |\lambda_1 &= 1.0 \\
\lambda_{2,3} &= (0.068, 0.7929) & |\lambda_{2,3}| &= 0.7958 \\
\lambda_{4,5} &= (-0.7357, 0.3023) & |\lambda_{4,5}| &= 0.7954 \\
\lambda_{6,7} &= (-0.7582, 0.2419) & |\lambda_{6,7}| &= 0.7959 \\
\lambda_8 &= -0.1974 & |\lambda_8| &= 0.1974.
\end{aligned}$$

Again, six out of eight eigenvalue are equal in magnitude.

The model parameters of this solution are $d_1 = 0.0252$, $d_2 = 0.3879$, $d_3 = 0.2858$, $\rho = -0.009$, $\beta = -0.2323$, $\gamma = -0.0294$, $m = 1.0$, $\alpha = 0.0757$, $d_x = 0.0024$, $d_y = 0.7901$, $d_z = 0.4323$, $l = 1.0$, $w = 0.3165$.

The corresponding trajectory has initial values of

$$x_0^T = (0.0876, -0.0114, -0.1731, 3.465, -0.0885, -0.0256, 0.3952, -0.3248)$$

and a cycle time of $T = 1.2314s$.

Again the matrix norms over one step of the robot are all larger than one:

$$\begin{aligned}
\sigma_{max} &= 4.066 \\
||C||_\infty &= 6.136 \\
||C||_1 &= 7.290.
\end{aligned}$$

The region of stability in which the robot can recover from perturbations is described by the stability margins

ϕ	+58%	-70%
ψ	+71%	-53%
θ_{st}	+6%	-5%
θ_{sw}	+0.5%	-0.9%
$\dot{\phi}$	+8%	-6%
$\dot{\psi}$	+20%	-15%
$\dot{\theta}_{st}$	+2%	-2%
$\dot{\theta}_{sw}$	+5%	-5%.

Figure 7.5 illustrates the differences between the original periodic trajectory and one with a perturbed initial value of ψ (-53%).

The robot also persists in its gait under the following perturbations of model parameter values:

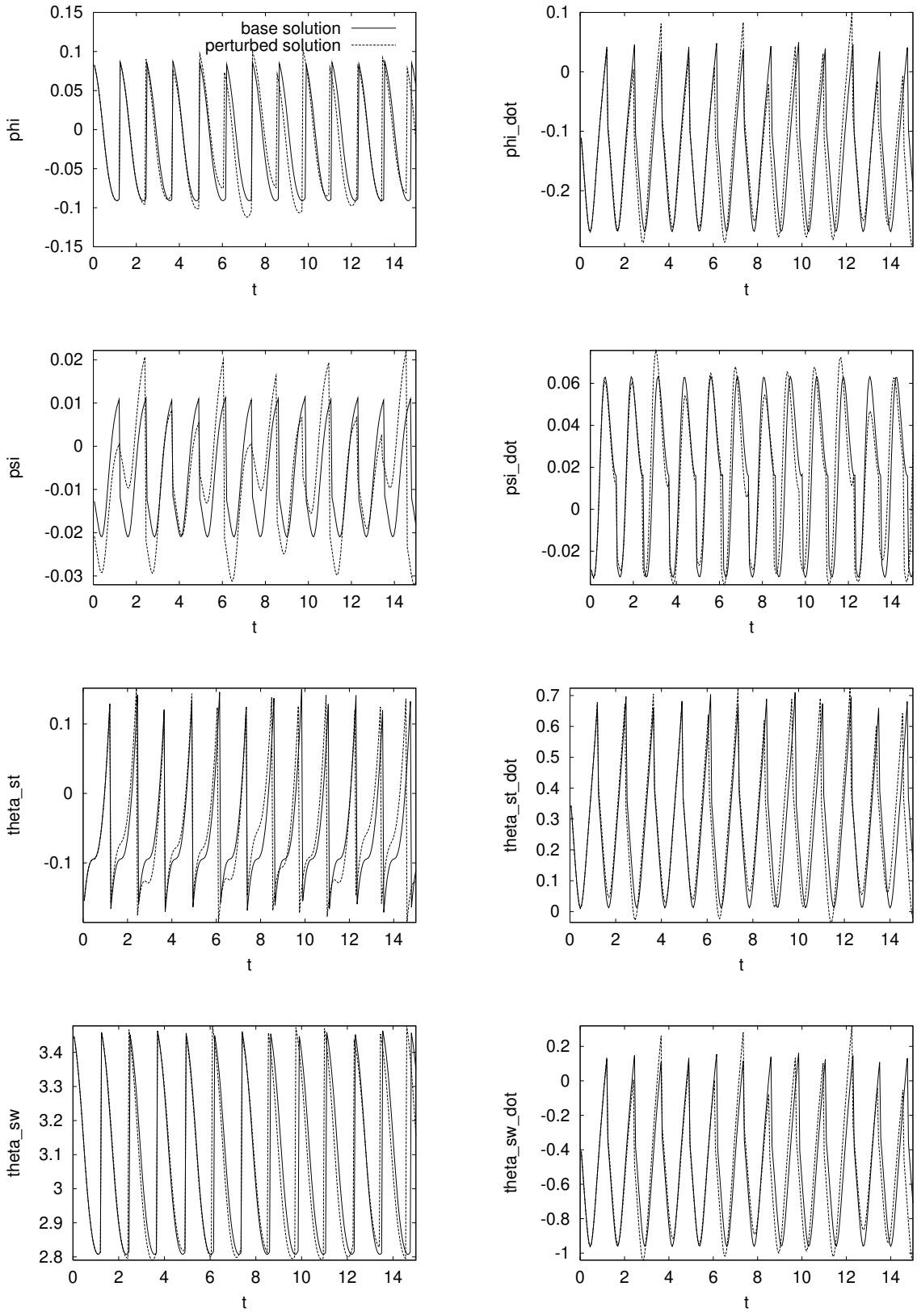


Figure 9.10: Most stable periodic solution for Tinkertoy with point feet with and without perturbation of ψ

d_1	+440%	-640%
d_2	+80%	-280%
d_3	+10%	-10%
ρ	+3200%	-3100%
β	+1500%	-1700%
γ	$+\infty$	$-\infty$
α	+13%	-36%
d_x	+130%	-120%
d_y	+9%	-9%
d_z	+8%	-6%
w	+13%	-6%.

9.3 Summary

The most important results of our computations for the Tinkertoy robot are:

- For the first time, stable solutions for passive-dynamic three-dimensional stiff-legged walking devices with point feet and toroidal feet have been presented.
- New improved solutions for the model version with disk feet have been found.
- All the solutions presented here are dynamically stable trajectories although the respective robots have no statically stable standing positions. In all cases the centers of mass lie above the centers of foot curvature: $z > r_1$ and $z > r_2$.
- In all three cases the stable solutions were not only characterized by small eigenvalues but also by large stability margins. We could again confirm that the existence of a contractive matrix norm is not necessary for excellent stability properties.
- Even for this passive system there is no solution with an induced matrix norm smaller than one. For the solutions found by eigenvalue optimization, 1-, 2-, or ∞-norm contract over a cycle of more than roughly twenty steps.
- Although we are aware that this is not a general rule for analytic matrices we have observed that for all the monodromy matrices of all robot models that we have studied during the research for this thesis, the maximum singular value was smaller than the 1- and ∞-norms and therefore closer to the spectral radius. This might be pure chance, but we assume that it is caused by the underlying dynamics.

Conclusions and Outlook

Summary & Conclusions

With this thesis, achievements have been made on both the engineering aspect of open-loop stable walking and running robots and the research about numerical methods necessary to find these solutions. The focus, however, has been set on the engineering side of the work and we therefore start with the description of these results that seem to be new in this field of research. We then proceed to demonstrate the achievements made on the numerical side of the work.

Walking Robots

Our main contributions lie in the field of theoretical walking robot research. Using optimization methods we were able to demonstrate the flexibility of the concept of open-loop control. It is applicable to a by far broader class of walking robots than was generally conceived before.

New Open-loop Stable Robot Models

During the course of our research we have discovered various robot configurations capable of stable motion without feedback:

- **Actuated 2D human-like walking robot:**
  This seems to be the first actuated open-loop stable robot mimicking human gait. It has knees and point feet and is powered by hip and knee torques. Only passive-dynamic systems of similar configuration have been known before.

- **Actuated 2D one-legged hopping robot:**
  Open-loop stable trajectories for hopping robots with point feet and circular feet have been found. Both versions have no statically stable standing configuration. Unlike previously assumed a circular foot is not necessary for open-loop stable hopping.

- **Passive-dynamic 3D walking robot:**
  For the first time, stable solutions for a 3D passive-dynamic three-dimensional stiff-legged walking robot with hip spacing and feet of point, disk, or toroidal shape have been computed. The solutions for all three versions correspond to statically unstable robot configurations. Stable behavior of a related physical robot had been observed before.

All stable solutions presented exhibit excellent linear stability properties since the maximum eigenvalues of their monodromy matrices are much smaller than one by magnitude. Nonlinear stability studies of the solutions in terms of allowable perturbations of initial values and model parameters have been performed. For two of our robots, the one-legged actuated hopping robot and the passive-dynamic walker, significant stability margins were computed. It is important to note that large stability margins can be achieved not only for passive but also for actuated systems. For the third robot, the actuated kneed walker, the stability margins are relatively small. A comparison with the corresponding passive walker studied in a previous publication shows that a lack of external periodic excitation facilitates the stabilization task considerably.

Modeling Periodic Gaits

Guidelines for the formulation of gaits as multi-phase periodic optimal control problems have been given including recommendations for the coordinate choice of the mechanical models. The order of motion phases should be prescribed. In order to obtain realistic motions, ground and joint impacts should be modeled with velocity discontinuities.

Understanding Human Walking

The kneed actuated walking robot can be considered as an abstract model of human gait. The 3D passive-dynamic robot also captures some features of human gait. The open-loop stability of these two models leads to the conjecture that humans might also be capable of stable walk without any sophisticated feedback.

Stability Optimization for General Dynamical Systems

Besides the immediate impacts on the applicational side, contributions of more general mathematical interest have been made.

Stability Optimization Procedure

We have developed a two-level optimization procedure for the improvement of open-loop stability. To our knowledge this is the first successful attempt to optimize the stability of

the solution of a periodic optimal control problem although many practical problems of this type exist. The method has two components:

- **Outer loop stability optimization**:
  Model design parameters are chosen as optimization variables. The monodromy matrix of the inner loop optimal control problem solution is computed and stability is measured terms of its spectral radius or an alternative objective function (see below). A direct search method is used for solution.

- **Inner loop periodic optimal control problem**:
  Controls, initial values and phase times are determined as solution to the periodic optimal control problem with some appropriate auxiliary objective function. Model parameters are fixed to the values given by the outer loop. The problem is solved by a direct method based on multiple shooting.

The applicability of this two-level procedure is not restricted to walking robots. If implemented along the schemes of our model library, any periodic dynamical system (with discontinuities and multiple phases) could be optimized by this approach. A natural split of variables into design and control variables leading to the two-level formulation is required.

Apparantly new formulas for derivatives of singular values with respect to matrix entries and for derivatives of monodromy matrices for discontinuous dynamics with respect to initial values and parameters have been derived.

Objective Functions for Stability Optimization

We have performed theoretical and numerical studies about the effects of several possible objective functions describing stability. The standard maximum eigenvalue criterion has produced the most stable results. The difficulties associated with this objective function have been discussed extensively. Depending on the algorithm chosen, alternative objective functions might be desirable.

For our systems we haven't found any solution for which the 1-, 2-, or ∞-norm of the monodromy matrix contract over the cycle of one step. Typically, contraction of perturbations occurs after several steps. Hence, the existence of a contractive norm of the monodromy matrix is not a necessary condition for excellent stability of a solution in the nonlinear sense. On the other hand, extremely large matrix norms seem to be one contributing factor – besides nonlinear effects – to small stability margins.

Using a norm of the monodromy matrix as optimization criterion did not produce stable solutions. With the optimal norms always being larger than one, the eigenvalues can and in our case did remain outside the unit circle. We even observed a deterioration of the spectral radius during the course of optimization. If for a specific application contraction of a norm over one cycle can be achieved this will of course also lead to stable solutions in

terms of the maximum eigenvalue. However we do not know, how to determine in advance if this is the case for some application.

Using the norm of a power of the monodromy matrix proved to be a much better choice. It helped to bring the eigenvalue into the stable region and delivered a solution that was very close to the solution found with eigenvalue optimization. There is no general rule for the choice of the matrix exponent.

We have found one surprising result which is not supported by standard matrix theory but might be caused by the underlying dynamics: the monodromy matrices of all robot models that we studied during our research had a maximum singular value that was smaller than the 1- and ∞-norms and therefore closer to the spectral radius. The maximum singular value might thus be preferred over some other norm as objective function.

Outlook

Based on the findings of our work described above we would like to note the following directions of research as particularly promising:

- **Biomechanical Applications:**
  Since human-like abstract robot models can be open-loop stable, a thorough study of the stability properties of human gait seems to be an interesting topic. A current field of research is the neuro-stimulation of paraplegic patients. Since those patients have only partial feedback at their disposal the concepts of open-loop control might be helpful in this case.

- **Manufacturing of designed robots:**
  Except for a variant of the 3D passive-dynamic walker none of the robots has been built yet. Manufacturing one of the open-loop controlled actuated robots would be interesting in order to demonstrate self-stabilizing properties in real-life experiments. One advantage of open-loop control is its simplicity of implementation since no sophisticated feedback control system is required.

- **Implementation of one-level approach & application:**
  The split of optimization variables that is required for the two-level approach and was straightforward for mechanical systems is not always possible for chemical processes. In the framework of SFB 359 the one-level approach to stability optimization also formulated in this theses will be implemented. Preliminary studies performed in this thesis on the choice of objective functions and on derivatives can be used for this purpose.

- **Combination with NMPC:**
  Open-loop stable systems are able to recover independently from the effects of small perturbations. In order to be able to also cope with more significant perturbations

an open-loop stable system could be equipped with additional nonlinear model predictive control (NMPC). The recent advances in this field make extremely short response times possible.

- **Optimization of robot gaits with respect to standard criteria:**
  There is a number of optimality questions that are of interest for the walking robot community, but have not yet been addressed, e.g. what is the fastest possible walking motion, and when does a mechanism start to run? With the gait models created in this thesis and the solution methods for the inner-loop optimal control problems, answering these questions is a simple and straightforward task. They have not been addressed in this thesis since they are not related to its central topic of stability.

Appendix A

Software Design and Implementation

The purpose of this appendix is to give some insight into the implementation of the numerical methods. It is not meant to give a complete overview of the developed software. At the beginning of our work it was not clear which types of algorithms we would finally use. We therefore had to choose an approach that allowed a large amount of flexibility and supported an exchange of components. Hence we have decided to use an object-oriented implementation in C++. See the classical book of Booch [12] for an introduction to object-oriented programming and e.g. Liberty [49] or Meyers [58], [59] for information on the C++ programming language.

A.1 Basic Software Components

A number of basic software components have been created:

- a mathematical base library containing different types of vector, matrix and tensor classes and standard operations
- an extension of the mathematical library for the computation of eigenvalues and singular values
- an optimization library with different types of functions, gradients etc., optimization problems, and optimization routines (see section A.1.1)
- extensions for eigenvalue and singular value optimization
- a model library allowing for the implementation of discontinuous multi-phase models
- interfaces to ODE-integrators
- integrator-model interface classes.

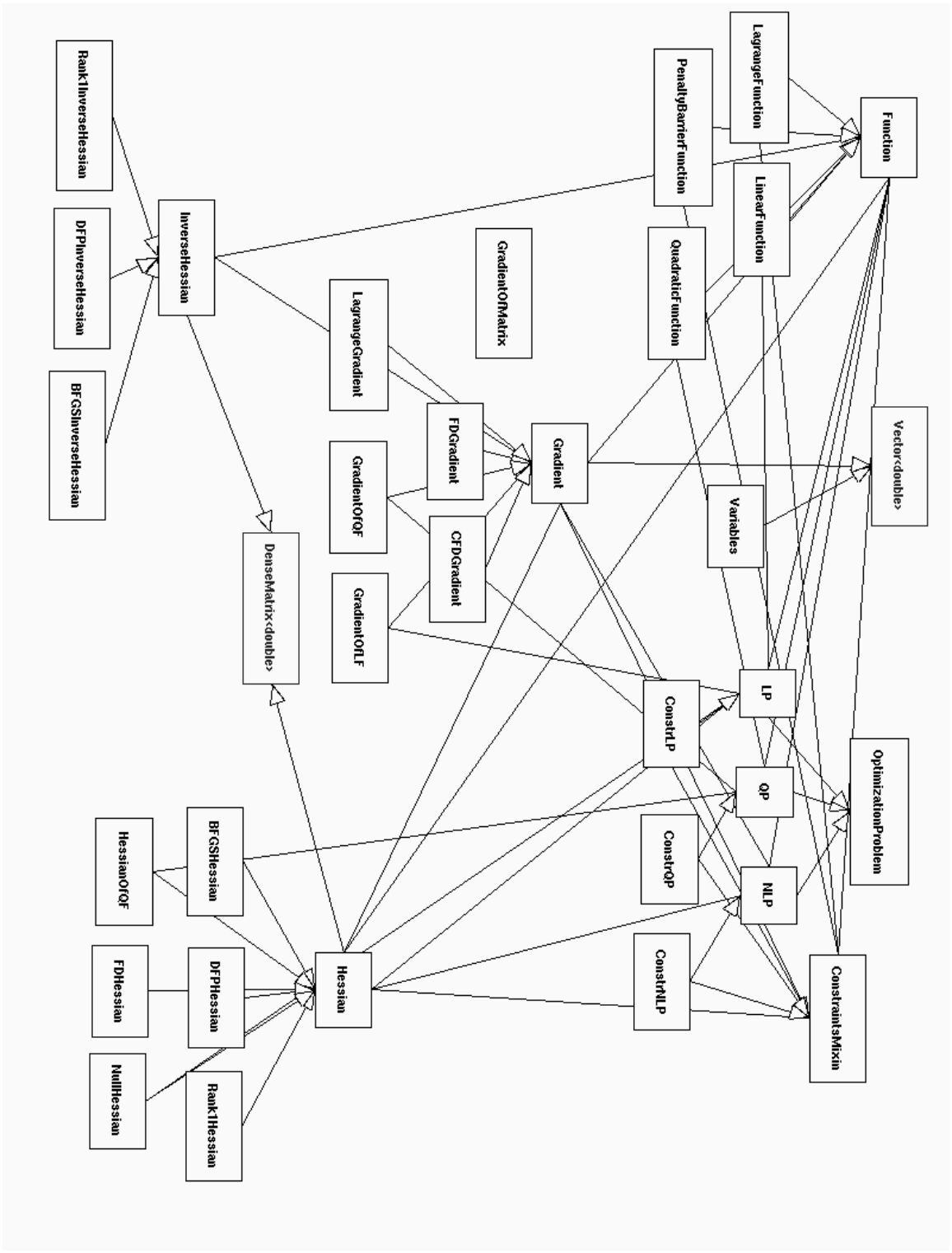


Figure A.1: Class hierarchy of optimization components and problem types (some of which are abstract data types)

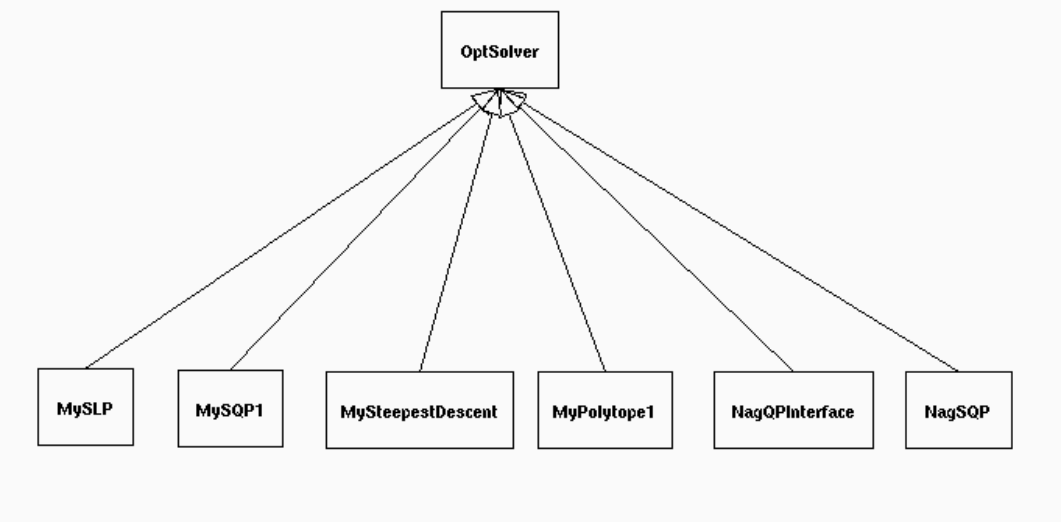


Figure A.2: Class hierarchy of optimization codes and interfaces

A.1.1 The Optimization Library

The optimization library consists of two parts. The first part contains optimization components like variables, functions, gradients, and Hessian matrices. Its hierarchy is shown in figure A.1. Arrows denote inheritance. Some of these classes inherit from base classes in the mathematical library, like *Variables* from *Vector*. The second part of the optimization library contains the optimization routines and interfaces (see figure A.2). All optimization routines are derived from the same base class and have the same interfaces. Optimization problems and solvers can be combined in a plug-and play manner where senseless combinations (like an NLP combined with and LP-solver) lead to an error message.

A.2 Two-level Stability Optimization Procedure

Figure A.3 illustrates the two-level stability optimization procedure that is a core component of this thesis. We show here the example of eigenvalue optimization for the Tinkertoy robot. The figure shows the hierarchy of classes as well as the integration of external components like MUSCOD-II for the solution of the inner-loop optimal control problem or LAPACK routine DGEEV for the computation of eigenvalues.

For the definition of a new stability optimization problem the following items have to be specified by the user:

- in main driver file *stabOpt++.cpp*:
  - model class, e.g. Tinkertoy(1);
  - name of inner loop optimal control problem, e.g. *tinkertoy1*

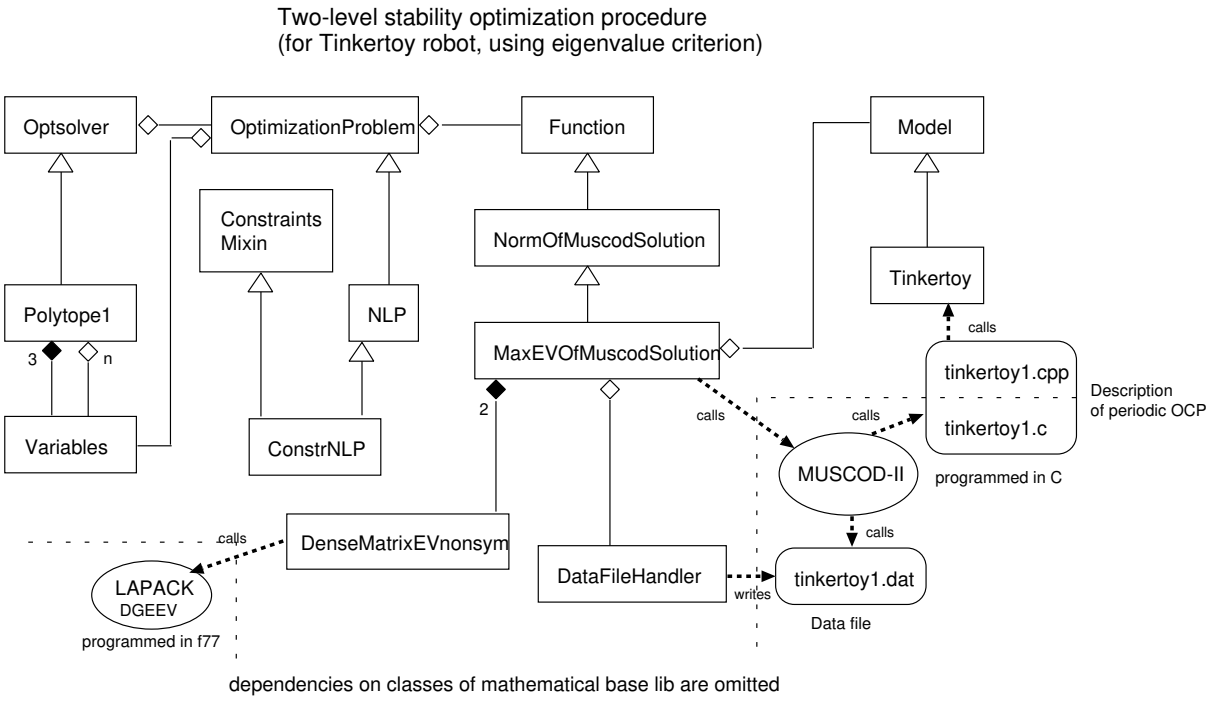


Figure A.3: Two-level stability optimization procedure

  - stability optimization criterion, e.g. maximum eigenvalue
  - constrained or unconstrained optimization problem

- in corresponding initialization file *stabOpt++.ini*:
  - model parameters to be modified
  - polytope unity length and scaling
- in optimal control problem source file, e.g. *tinkertoy1.c* & *tinkertoy1.cpp*:
  - inner loop objective function
  - coupled and decoupled multipoint constraints
- in problem data file, e.g. *tinkertoy1.dat*:
  - start values for parameters, state variables at multiple shooting points, controls and phase times
  - corresponding bounds
  - number of multiple shooting intervals, control discretization types, phase types, switching structure
- implement new model class if not yet available in library.

The following listing gives the different objective functions for stability optimization: maximum eigenvalue, maximum singular value, ∞-norm. They are all derived from the same base class *NormOfMuscodSolution*:

```
class NormOfMuscodSolution : public Function{

 protected:

  Model *p_TheModel;
  char *p_problemName;
  DataFileHandler *p_MuscodData;
  ifstream MuscodResultFile;
  Vector<int> OptParams;
  int noOfPhases;
  int stateDim;
  int paramDim;
  int controlDim;
  Vector<double> States;
  Vector<double> Params;
  Vector<double> Controls;
  Vector<double> XdotPlus;
  Vector<double> XdotMinus;
  Vector<double> dSWdX;
  Vector<double> dSWdP;
  Vector<double> dSWdU;
  double dSWdT;
  double swDot;
  Vector<double> PhaseTimes;
  DenseMatrix<double> *p_phaseTransferMatrix;
  DenseMatrixEVnonsym<double> JacPoincareMapEV;
  DenseMatrixEVnonsym<double> ProjectedJacPoincareMapEV;
  DenseMatrix<double> LocalMatrix;
  int updateIniFlag;
  void calcJacPoincareMap();
  void calcProjectedJacMap();
  void writeLogFile();


 public:
  NormOfMuscodSolution(Model *p_Model, char *p_pbName,
       DataFileHandler *p_DataFile, char *p_resFileName,
       Vector<int>& WhichParams);
  virtual  ~NormOfMuscodSolution();
  void setIniUpdate();
  virtual double calculate(Variables& X) = 0;

};


class MaxEVOfMuscodSolution : public NormOfMuscodSolution{

 protected:
  double maxEV;
```

```
  void writeLogFile();

 public:
  MaxEVOfMuscodSolution(Model *p_Model, char *p_pbName,
DataFileHandler *p_DataFile, char *p_resFileName,
Vector<int>& WhichParams);
  virtual  ~MaxEVOfMuscodSolution();
  virtual double calculate(Variables& X);

};

class MaxSVOfMuscodSolution : public NormOfMuscodSolution{

 protected:
  double  maxSV;
  int matrixPower;
  DenseMatrixSV<double> PoincareMapSV;
  DenseMatrixSV<double> ProjectedPoincareMapSV;
  void writeLogFile();

 public:
  MaxSVOfMuscodSolution(Model *p_Model, char *p_pbName,
DataFileHandler *p_DataFile, char *p_resFileName,
Vector<int>& WhichParams, int power = 1);
  virtual  ~MaxSVOfMuscodSolution();
  virtual double calculate(Variables& X);

};


class InfNormOfMuscodSolution : public NormOfMuscodSolution{

 protected:
  double infNorm;
  int matrixPower;
   void writeLogFile();

 public:
  InfNormOfMuscodSolution(Model *p_Model, char *p_pbName,
DataFileHandler *p_DataFile, char *p_resFileName,
Vector<int>& WhichParams, int power = 1);
  virtual  ~InfNormOfMuscodSolution();
  virtual double calculate(Variables& X);

};
```

A.3 Determination of Stability Margins

In order to determine stability margins of a stable periodic solution the following steps are necessary:

- in main driver file *sim++.cpp*:
  - specify model
  - indicate path of file containing optimization results with trajectory information
  - give optimization data file (discretization data is needed for interpretation of previous file)
- in initialization file *sim++.ini*:
  - set integration start (default $t_s = 0$) and end times
  - (select integrator output mode – step or continuous)
  - modify respective perturbation factors of initial values or parameters
- start integration and check results.

Bibliography

[1] J. Adolfsson, H. Dankowicz, and A. Nordmark. 3-D stable gait in passive bipedal mechanisms. In *Biology and Technology of Walking*, pages 253 – 259. Euromech Colloquium 375, Munich, 1998.

[2] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton. Primal-dual interior point methods for semidefinite programming: Convergence rates, stability, and numerical results. *SIAM Journal on Optimization*, 8(3):746 – 768, 1998.

[3] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostruchov, and D. Sorensen. *Lapack Users' Guide*. SIAM, 1995.

[4] U. M. Ascher, R. M. M. Mattheij, and R. D. Russel. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*. Prentice Hall, 1998.

[5] G. L. Baker and J. P. Gollub. *Chaotic Dynamics*. Cambridge University Press, Cambridge, UK, 1990.

[6] I. Bauer, H. G. Bock, and J. P. Schlöder. DAESOL – a BDF-code for the numerical solution of differential equations. Technical report, IWR, University of Heidelberg, SFB 359, 1999.

[7] K. Berns. Walking machines catalogue. *http://www.fzi.de/ipt/WMC/walking-machines-katalog/walking-machines-katalog.html*, 2000.

[8] H. G. Bock. Zur numerischen Behandlung zustandsbeschränkter Steuerungsprobleme mit Mehrzielmethode und Homotopieverfahren. *Zeitschrift für Angewandte Mathematik und Mechanik (ZAMM)*, 57:T266 – T268, 1977.

[9] H. G. Bock. Numerische Berechnung zustandsbeschränkter optimaler Steuerungen mit der Mehrzielmethode. Technical report, Carl-Cranz-Gesellschaft, 1978.

[10] H. G. Bock. *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*. PhD thesis, Universität Bonn, 1985.

[11] H. G. Bock and K.-J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *Proceedings of the 9th IFAC World Congress, Budapest*, pages 242–247. International Federation of Automatic Control, 1984.

[12] G. Booch. *Objektorientierte Analyse und Design.* Addison Wesley, 1994.

[13] I. Bronstein, K. Semendjajew, G. Musiol, and H. Mühlig. *Taschenbuch der Mathematik.* Verlag Harri Deutsch, Thun and Frankfurt am Main, 1997.

[14] P. H. Channon, S. H. Hopkins, and D. T. Pham. Derivation of optimal walking motions for a bipedal walking robot. *Robotica*, 10:165 – 172, 1992.

[15] M.-Y. Cheng and C.-S. Lin. Measurement of robustness for biped locomotion using a linearized Poincar'e map. *Robotica*, 14:253 – 259, 1996.

[16] F. H. Clarke. *Optimization and Nonsmooth Analysis.* Number 5 in Classics in Applied Mathematics. SIAM, 1990.

[17] Honda Motor Co. Humanoid robot. *http://world.honda.com/robot*, 2001.

[18] M. J. Coleman. *A stability study of a three-dimensional passive-dynamic model of human gait.* PhD thesis, Cornell University, February 1998.

[19] M. J. Coleman, M. Garcia, K. D. Mombaur, and A. Ruina. Prediction of stable walking for a toy that cannot stand. *Physical Review E*, to appear 2001.

[20] M. J. Coleman and A. Ruina. An uncontrolled walking toy that cannot stand still. *Physical Review Letters*, 80(16):3658 – 3661, April 1998.

[21] F. Colonius. *Optimal Periodic Control.* Springer, 1980.

[22] J. Cronin. *Differential Equations – Introduction and Qualitative Theory.* Marcel Dekker, 1994.

[23] H. de Man, D. Lefeber, and J. Vermeulen. Design and control of a one-legged robot hopping in irregular terrain. In *Biology and Technology of Walking*, pages 173–180. Euromech Colloquium 375, Munich, 1998.

[24] J. W. Demmel. *Applied Numerical Algebra.* SIAM, 1997.

[25] D. Dinkler. Dynamik I und II. Vorlesungsmanuskripte, ISD, Universtät Stuttgart, 1992.

[26] D. Dinkler. Aeroelastik. Vorlesungsmanuskript, ISD, Universtät Stuttgart, 1993.

[27] R. Eppler. Technische Mechanik II. Manuskript zur Vorlesung, IAM, Universtät Stuttgart, 1989.

[28] R. Fletcher. *Practical Methods of Optimization.* Wiley, 1987.

[29] O. Föllinger. *Regelungstechnik.* Hüthig, 1990.

[30] M. Garcia. *Stability, Scaling, and Chaos in Passive-Dynamic Gait Models.* PhD thesis, Cornell University, January 1999.

[31] M. Garcia, A. Chatterjee, and A. Ruina. Speed, efficiency, and stability of small-slope 2-D passive dynamic walking. In *Proceedings of IEEE International Conference on Robotics and Automation*, Leuven, Belgium, 1998.

[32] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization.* Academic Press, 1981.

[33] C. J. Goh and K. L. Teo. On minmax eigenvalue problems via constrained optimization. *Journal of Optimization Theory and Applications*, 57(1):59 – 68, 1988.

[34] S. Goyal. Second order kinematic constraint between two bodies rolling, twisting, and slipping against each other while maintaining point contact. Technical Report TR 89-1043, Department of Computer Science, Cornell University, 1989.

[35] D. T. Greenwood. *Principles of Dynamics.* Prentice-Hall, 1988.

[36] Ch. Großmann and J. Terno. *Numerik der Optimierung.* Teubner, Stuttgart, 1993.

[37] M. Hartel. *Numerische Berechnung periodischer Strategien zur optimalen Prozeßsteuerung.* PhD thesis, Universität Heidelberg, 1996.

[38] W. Hauger, W. Schnell, and D. Gross. *Technische Mechanik, Band 3 – Kinetik.* Springer, 1995.

[39] C. Helmberg and F. Rendl. A spectral bundle method for semi-definite programming. *SIAM Journal on Optimization*, to appear. ZIB Preprint, SC-97-37.

[40] J. C. Hsu and A. U. Meyer. *Modern Control Principles and Applications.* McGraw-Hill, 1968.

[41] M. Hubbard. Lateral dynamics and stability of the skateboard. *Transactions of the ASME*, 46:931 – 936, 1979.

[42] Y. Hurmuzlu. Dynamics of bipedal gait: Part II – Stability analysis of a planar five-link biped. *Journal of Applied Mechanics*, 60:337 – 343, june 1993.

[43] T. Karčnik, A. Kralj, and T. Bajd. Crutch supported locomotion as a quadrupedal gait pattern. In *Biology and Technology of Walking*, pages 74 – 81. Euromech Colloquium 375, Munich, 1998.

[44] Y. A. Kuznetsov. *Elements of Applied Bifurcation Theory.* Springer, 1998.

[45] MIT Leg Lab. Leg lab robots. *http://www.ai.mit.edu/projects/leglab/robots/robots-main.html*, 2001.

[46] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM Journal on Optimization*, 9(1):112 – 147, 1998.

[47] D. B. Leineweber. Analyse und Restrukturierung eines Verfahrens zur direkten Lösung von Optimal-Steuerungsproblemen (The Theory of MUSCOD in a Nutshell). Technical report, IWR PPreprint, 1996.

[48] D. B. Leineweber. *Efficient Reduced SQP Methods for the Optimization of Chemical Processes Described by Large Sparse DAE Models*, volume 3 of *Fortschrittsberichte*. VDI, 1999.

[49] J. Liberty. *Teach Yourself C++ in 21 Days*. Sams Publishing, 1998.

[50] T. McGeer. Stability and control of two-dimensional biped walking. Technical Report CSS-IS TR 88-01, Simon Fraser University, 1988.

[51] T. McGeer. Passive bipedal running. *Proceedings of the Royal Society of London*, B 240:107 – 134, 1990.

[52] T. McGeer. Passive dynamic walking. *International Journal of Robotics Research*, 9:62–82, 1990.

[53] T. McGeer. Passive dynamic biped catalogue. In R. Chatila and G. Hirzinger, editors, *Proceedings of the 2nd International Symposium of Experimental Robotics, Toulouse*. Springer-Verlag, New York, 1991.

[54] T. McGeer. Principles of walking and running. In *Advances in Comparative and Environmental Physiology*. Springer-Verlag, Berlin, 1992.

[55] K. I. M. McKinnon. Convergence of the nelder-mead simplex method to a nonstationary point. *SIAM Journal on Optimization*, 9(1):148 – 158, 1998.

[56] L. Meirovitch. *Methods of Analytical Dynamics*. McGraw-Hill, 1970.

[57] K. D. Metzger. Einsatz eines dynamischen Optimierungsverfahrens zur Muskelkraftberechnung menschlicher Bewegungen. Master's thesis, Institut A für Mechanik, Universität Stuttgart, 1995.

[58] S. Meyers. *Effective C++*. Addison-Wesley, 1998.

[59] S. Meyers. *More Effective C++*. Addison-Wesley, 1998.

[60] S. Mochon and T. S. McMahon. Ballistic walking. *Biomechanics*, 13:49 – 57, 1980.

[61] S. Mochon and T. S. McMahon. Ballistic walking: An improved model. *Mathematical Biosciences*, 52:241 – 260, 1980.

[62] K. D. Mombaur, H. G. Bock, and R. W. Longman. Stable, unstable, and chaotic motions of bipedal walking robots without feedback. In F. L. Chernousko and A. L. Fradkov, editors, *Proceedings of 2nd International Conference on Control of Oscillations and Chaos, St. Petersburg*, volume 2, pages 282 – 285, 2000.

[63] K. D. Mombaur, H. G. Bock, and J. P. Schlöder. Numerical generation and stabilization of periodic gaits. Technical Report 2000-39, IWR Preprint, Universität Heidelberg, 2000.

[64] K. D. Mombaur, H. G. Bock, J. P. Schlöder, and R. W. Longman. Human-like actuated walking that is asymptotically stable without feedback. In *Proceedings of IEEE International Conference on Robotics and Automation*, 2001.

[65] K. D. Mombaur, H. G. Bock, J. P. Schlöder, M. J. Winckler, and R. W. Longman. Open-loop stable control of running robots - a numerical method for studying stability in the context of optimal control problems. In *Proceedings of Clawar '98, Brussels*, pages 89 – 94, 1998.

[66] J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308 – 313, 1965.

[67] S. Otterbein. Stabilisierung des n-Pendels und der Indische Seiltrick. *Archive for Rational Mechanics and Analysis*, 78(4):381 – 393, 1982.

[68] M. L. Overton. On minimizing the maximum eigenvalue of a symmetric matrix. *SIAM Journal on Matrix Analysis and Applications*, 9:256 – 268, 1988.

[69] M. L. Overton. Large-scale optimization of eigenvalues. *SIAM Journal on Optimization*, 2(1):88–120, 1992.

[70] M. L. Overton and R. S. Womersley. On minimzing the spectral radius of a nonsymmetric matrix function: Optimality conditions and duality theory. *SIAM Journal on Matrix Analysis and Applications*, 9(4):473 – 498, 1988.

[71] M. L. Overton and R. S. Womersley. Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices. *Mathematical Programming*, 62:321 – 357, 1993.

[72] E. R. Panier. On the need for special purpose algorithms for minimax eigenvalue problems. *Journal of Optimization Theory and Applications*, 72(2):279 – 287, 1989.

[73] G. A. Pratt and M. M. Williamson. Series Elastic Actuators. In *Proceedings of IROS*, Pittsburgh, 1995.

[74] J. E. Pratt. *Exploiting Inherent Robustness and Natural Dynamics in the Control of Bipedal Walking Robots*. PhD thesis, Massachusetts Institute of Technology, 2000.

[75] M. H. Raibert and I. E. Sutherland. Machines that walk. *Scientific American*, 248(1):32 – 41, Jan. 1983.

[76] R. P. Ringrose. Self-stabilizing running. Technical report, Massachusetts Institute of Technology.

[77] R. P. Ringrose. *Self-Stabilizing Running.* PhD thesis, Massachusetts Institute of Technology, 1997.

[78] A. Ruina. Non-holonomic stability aspects of piecewise-holonomic systems. *Reports on Mathematical Physics*, 1998.

[79] A. Ruina. Human Power, Biomechanics and Robotics Lab, Cornell University. *www.tam.cornell.edu/ ruina/hplab*, 2001.

[80] J. La Salle and S. Lefschetz. *Stability by Liapunov's Direct Method.* Academic Press, 1961.

[81] S. Schaal and C. G. Atkeson. Open loop stable control strategies for robot juggling. In *IEEE International Conference on Robotics and Automation*, pages 913 – 918, 1993.

[82] H. R. Schwarz. *Numerische Mathematik.* Teubner, 1997.

[83] A. Shapiro and M. K. H. Fan. On eigenvalue optimization. *SIAM Journal on Optimization*, 5(3):552 – 569, 1995.

[84] J. Simon. Modellierung von Kontaktereignissen bei der Simulation von Laufvorgängen. Master's thesis, Universität Heidelberg, 1998.

[85] J. Stoer. *Numerische Mathematik 1.* Springer, 1994.

[86] J. Stoer and R. Bulirsch. *Numerische Mathematik 2.* Springer, 1990.

[87] T. Stoßmeister. PhD thesis, Universtät Heidelberg, to appear 2001.

[88] C. M. Thompson and M. H. Raibert. Passive dynamic running. In V. Hayward and O. Khatib, editors, *Proceedings of International Symposium of Experimental Robotics*, pages 74 – 83. Springer-Verlag, New York, 1989.

[89] M. W. Thring. *Robots and Telechirs.* Ellis Horwood, 1983.

[90] V. Torczon. *Multi-Directional Search: A Direct Search Algorithm for Parallel Machines.* PhD thesis, Rice University, Houston, Texas, May 1989.

[91] V. Torczon. On the convergence properties of pattern search algorithms. *SIAM Journal on Optimization*, 7(1):1 – 25, 1997.

[92] L. N. Trefethen and D. Bau III. *Numerical Linear Algebra.* SIAM, 1997.

[93] R. v. Schwerin. *Numerical Methods, Algorithms and Software for Higher Index Nonlinear Differential-Algebraic Equations in Multibody System Simulation.* PhD thesis, Universität Heidelberg, 1997.

[94] R. v. Schwerin and M. J. Winckler. Anleitung für das ODE-Integratorpaket ODESIM, Version 0.97a. Technical report, IWR, 1994.

[95] R. v. Schwerin and M. J. Winckler. A guide to the integrator library MBSSIM, version 1.00. Technical Report 94 - 75, IWR Preprint, 1994.

[96] R. v. Schwerin, M. J. Winckler, and V. H. Schulz. Parameter estimation in discontinuous descriptor models. In D. Bestle and W. Schiehlen, editors, *IUTAM Symposium on Optimization of Mechanical Systems*, pages 269–276. Kluwer Academic Publishers, 1996.

[97] M. Vukobratovic, B. Borovac, D. Surla, and D. Stokic. *Scientific Fundamentals of Robotics 7: Biped Locomotion - Dynamics, Stability, Control and Applications*. Springer, 1990.

[98] W. Walter. *Gewöhnliche Differentialgleichungen*. Springer, 1996.

[99] T. E. Wei, G. M. Nelson, R. D. Quinn, H. Verma, and S. L. Garverick. Design of a 5-cm monopod hopping robot. In *Proceedings of IEEE International Conference on Robotics and Automation*, 2000. http://www.fluggart.cwru.edu/icra2000/icra2000.html.

[100] H. Werner and H. Arndt. *Gewöhnliche Differentialgleichungen*. Springer, 1986.

[101] J. H. Wilkinson. *The Algebraic Eigenvalue Problem*. Clarendon Press, Oxford, 1965.

[102] M. Winckler. *Numerische Werkzeuge zur Simulation, Visualisierung und Optimierung unstetiger dynamischer Systeme*. PhD thesis, Universtät Heidelberg, 2000.

[103] M. J. Winckler and J. Huber. JAFV – An OpenGL visualization testbed for dynamic models. Technical report, IWR Preprint, 1999.

[104] *Proceedings of Clawar 98, First International Conference on Walking and Climbing Robots*, Brussels, 1998.

[105] *Proceedings of ICAR 97, 8th International Conference on Advanced Robotics*, Workshop II : New Approaches in Dynamic Walking and Climbing Machines, Monterey, 1997.